DOCUMENT RESUME

ED 262 266 CE 042 702

AUTHOR	Auer, Herbert J.
TITLE	Robotics: Instructional Manual. The North Dakota High Technology Mobile Laboratory Project.
INSTITUTION	North Dakota Univ., Grand Forks. Dept. of Industrial Technology.
SPONS AGENCY	North Dakota State Board for Vocational Education, Bismarck.
PUB DATE	85
NOTE	246p.; For related documents, see CE 042 701-705.
PUB TYPE	Guides - Classroom Use - Guides (For Teachers) (052) -- Computer Programs (101)
EDRS PRICE	MF01/PC10 Plus Postage.
DESCRIPTORS	Competency Based Education; *Computer Oriented Programs; Course Content; Equipment; Industrial Arts; Learning Activities; Lesson Plans; *Microcomputers; Postsecondary Education; *Robotics; Secondary Education; Teaching Methods; Vocational Education; Workshops

ABSTRACT

This instructional manual contains 20 learning activity packets for use in a workshop on robotics. The lessons cover the following topics: safety considerations in robotics; introduction to technology-level and coordinate-systems categories; the teach pendant (a hand-held computer, usually attached to the robot controller, with which the operator can guide the robot through a series of points or in a motion pattern); the robot's main structure; robotics maintenance and troubleshooting; robot waist, bicep, and elbow actions; robot wrist flex and rotate actions; robotic end-effectors and actions; the robot's working envelope; the "hard" and "soft" home positions/movements; teach pendant: the "LEARN" A sequence mode; teach pendant: "EDITing" a LEARNed program; the robot's drive train and drive-motor steps; robotics feedback, optical encoders, and drive-motors; robotics end-effector placement specifications; robot pick-and-place operations; robot point-to-point operations; robot continuous path operations; robot program loading; robotics languages; and the future of robotics. Each learning activity packet lists prerequisites, purpose, and competencies to be developed and contains information sheets, illustrations, and learning activities. A lengthy appendix to the instructor's manual explains the XR series teach pendant. (KC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

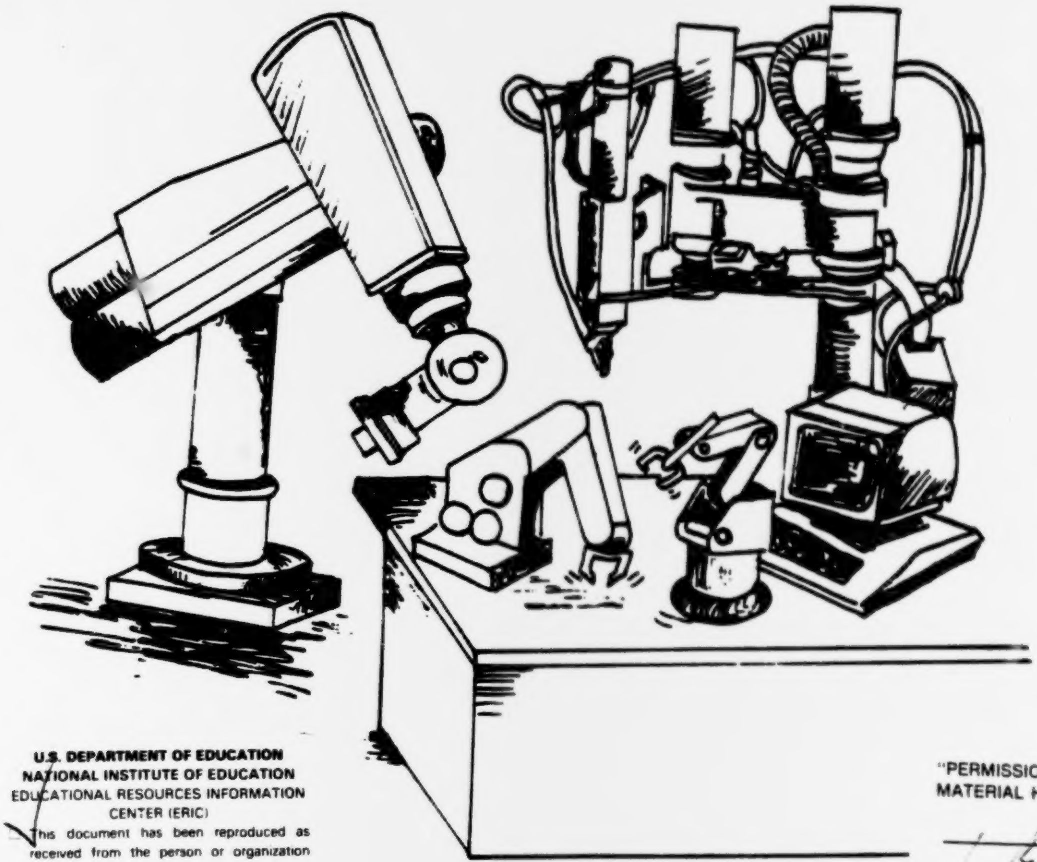
ED262266

BEST COPY AVAILABLE

ROBOTICS:

Instructional Manual

Writer: Dr. Herbert J. Auer



U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

● Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)."

CE042 702

The Industrial Technology Department, University of North Dakota --
High Technology Mobile Laboratory Project Funded by the North Dakota State Board
For Vocational Education -- Mr. Jerry Balistreri and Dr. Myron Bender, Co-Directors.

1985

The North Dakota
High Technology Mobile Laboratory Project

Conducted Through
The Department of Industrial Technology
University of North Dakota
Grand Forks 58202

Funded By
The North Dakota State Board For
Vocational Education
State Capitol
Bismarck, North Dakota

Project Co-Directors:
*Dr. Myron Bender
Department of Industrial Technology
*Mr. Jerry Balistreri
Former North Dakota State
Industrial Arts Supervisor

Specialty Consultant & Writers:
*Dr. Herbert J. Auer
Robotics Instructional Manual
Photovoltaic Instructional Manual
*Dr. Wan-Lee Cheng
Computer-Aided Design
Instructional Manual
*Dr. John W. Sinn
Computer Numerical Control Lathe
Instructional Manual
*Dr. Luvern R. Eickhoff
Laser and Fiber Optics
Instructional Manual

ROBOTICS

INSTRUCTIONAL LABORATORY MANUAL

TABLE OF CONTENTS

NO.	TITLE	PREREQUISITE(S)
1	Safety Considerations In Robotics	None
2	Introduction: Technology-Level Categories	1
3	Introduction: Coordinate-Systems Categories	1 & 2
4	The Teach Pendant: Connections, Power-Up and Simple Operations	1 & 3
5	The Robot's Main Structure	4
6	Robotics Maintenance and Troubleshooting	5
7	Teach Pendant: Robot Waist, Bicep and Elbow Actions	4 & 5
8	Teach Pendant: Robot Wrist Flex and Rotate Actions	4 & 5
9	Robotic End-Effectors and Actions	4 & 5
10	The Robot's Working Envelope	4 & 5
11	The "Hard" and "Soft" Home Positions/Movements	4
*12	Teach Pendant: The "LEARN" A Sequence Mode	4 & 11
*13	Teach Pendant: "EDITing" A LEARNed Program	4, 11 & 12
14	The Robot's Drive Train and Drive-Motor Steps	4 & 11
15	Robotics Feedback, Optical Encoders and Drive-Motors	4, 12 & 13
16	Robotics End-Effector Placement Specifications	4, 11 & 12
**17	Robot Pick-And-Place (PNP) Operations	4, 11, 12 & 13
**18	Robot Point-To-Point (PTP) Operations	4, 11, 12 & 13
**19	Robot Continuous Path (CP) Operations Demonstration	4, 11, 12 & 13
20-A	Robot Program Up-Loading/Down-Loading To/From A Computer	17, 18 or 19
20-B	Robotics Languages	2
20-C	The Future of Robotics	2

Appendix A: XR SERIES TEACH PENDANT (Manual)

* It is suggested that these experiments be done in conjunction with one another.

** The programs developed for these experiments may be worth saving on disk, as described in No. 20-A.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 1

LAP TITLE: Safety Considerations In Robotics

PREREQUISITE LAP (S) : None

LAP PURPOSE: To provide information regarding potential hazards, safety considerations, and safe practices in the use of robots.

COMPETENCIES TO BE DEVELOPED:

1. Define and recognize various potential hazards and the safety considerations associated with such hazards.
2. Be able to practice safe attitudinal and operational habits when working with the robot.

BACKGROUND INFORMATION

The safety considerations associated with the robotics field are numerous and very important. These considerations involve both the robot and other nearby related machines and the safety of all persons who may be within the vicinity of the robot and related machines.

Safety Objectives

The objectives for any robot application should include:

1. Providing safe working conditions for all persons in the entire working area surrounding the robot.
2. Provide proper supportive controls for the robot and related machines to prevent dammage to the robot itself or to objects which it handles.
3. Provide adequate safety controls on all equipment so that if a power failure occurs the entire system is rendered safe and actions cannot resume unless complete safety is assured.

Safety Requirements

The requirements for safe operation of robots typically include:

1. Proper design and assembly/connection of all mechanical and electrical equipment within the working region of the robot.
2. Fail-safe behavior of all equipment. Fail-safe means that if failure of any or all machines should occur, those machines will go to the most safe condition, position, or location possible. For some machines this means that their movement(s) need to continue to the safe position through use or auxiliary (stored) power. For other machines the fail-safe condition may involve automatic setting of brakes when a failure occurs.
3. Provide proper safety controls so that only authorized personnel can restart the system, and yet the system will shut down if a human error should occur within the system's work space.

Safety Devices

The safety devices often used with robots are not unlike those found on, near, or surrounding other automatic machines in industries. These devices may include the following:

1. Proper insulation/shielding of electrical cables or hydraulic (oil) and pneumatic (air) lines.
2. Proper shielding of mechanical drives, such as gears, chains and sprokets, rotating shafts, etc.

3. Use of curtains or shields around welders, lathes, milling machines, etc., to protect operators or other people in the work region.
4. Use of a fence surrounding the robot's work region with "windows" for parts entry and retrival, but also with controls which will stop the robot and/or other machines if entry occurs through an unsafe area.
5. Use of an infrared light curtain surrounding the robot. Such a curtain includes an infrared light source to provide the curtain and infrared light detectors to sense the presence or absence of the light. An infrared light curtain can be programmed to provide the desired entry/retrival "windows", and it is uneffected by visible light. An illustration of an infrared light curtain is shown in Figure 1.

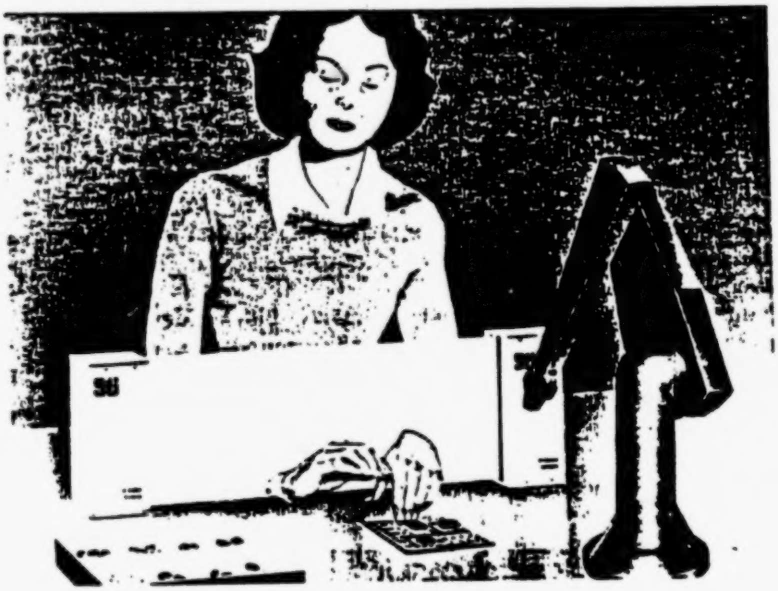


Figure 1. Miniaturized programmable light curtain. Courtesy of Scientific Technology, Inc., Mountain View, Calif. SOURCE: Edward Kafrissen and Mark Stephans, Industrial Robots and Robotics (Reston: Reston Publishing Co., 1984), p.149, figure 5.27.

LEARNING ACTIVITIES

The robot furnished within the High Technology Mobile Laboratory has several safety controls built into its Teach Pendant. These are described in the below listed pages of the XR SERIES TEACH PENDANT, Version 2.0 manual located in Appendix A in the back of this notebook. Look up and read these small sections of the manual to answer the following:

1. The RESET button is located on the bottom right edge of the Teach Pendant. The robot will stop anytime this button is pressed. However, it may not be the desired way to stop the robot because all memory is lost when it is pressed. See paragraph 1, Section 7.1 on page 11 of the manual. What does pressing the RESET button do to the memory?

2. The HOLD Function (not a key or button on the Teach Pendant) is described in #2 on page iii of the manual. What will the HOLD function do?

3. The STALL Routine is a rather complex safety feature which is fully described in Secion 10 on page 27 of the manual. Under what condition(s) will a STALL occur (see paragraph 1)?

4. The "100" step key (located second-to-left in the bottom row of the Teach Pendant) has a brief description in paragraph 3 of Section 6.2 on page 10 of the manual. Describe the possible danger in using this key at the wrong time.

Other safety considerations of particular concern for using the robot furnished in the High Technology Mobile Laboratory are as follows:

1. The robot has open (unguarded) chains, sprockets, pulleys, and arm parts. Describe the hazards which these represent and what precautions these require of you when using the robot.

____

____

2. The robot's arm can be extended to near the end of the "+x" axis which is drawn on the grid table. Describe what precautions this requires of you when swinging the robot's arm in the "+y" and "-y" directions. ____

____

____

____

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 2

LAP TITLE: Introduction: Technology-Level Categories

PREREQUISITE LAP (S) : 1

LAP PURPOSE: To provide background information regarding definitions, capabilities and potential uses of the various technology-level categories of robots.

COMPETENCIES TO BE DEVELOPED:

1. Define various basic terms associated with the field of robotics.
2. Identify capabilities and/or potential uses for the low-, medium- and high-technology level categories of robots.

BACKGROUND INFORMATION

Definitions

Following is a list of general terms and definitions regarding industrial robots and the field of robotics. Other terms will be introduced as needed in later Learning Activity Packets (LAPs).

Robot - A reprogrammable, multi-functional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks in various industries.

Low-Technology Robots - They are usually restricted to three axes of rectilinear or cylindrical movement, are logic system controlled, and regulated by adjustable stops.

Medium-Technology Robots - They are primarily of a sphereical or human-like arm design with five axes of movement, are microcomputer controlled, and regulated by a feedback system.

High-Technology Robots - They typically have six axes of human-like arm movement, are microprocessor controlled, and are often interfaced with external sensors.

Axis - An imaginary line around which an object or component rotates.

Actuator - A device which converts electrical, hydraulic, or pneumatic energy into mechanical linear or rotary motion.

End Effector - The tool or holding device mounted at the end of the manipulator arm.

Gripper - The 'hand' of a robot, which is a type of end effector usually modeled after the human hand.

Low- Medium- and High-Technology Robot Capabilities

Figure 1 shows the three categories of industrial robots: low-technology, medium-technology, and high-technology. Each robot category has its own unique characteristics, such as: number of axes of motion, type of controller, method of controller programming, type of manipulator motion, and kind of actuator used. Under the general classification of industrial robots, these categories can be based on the type of work the robot is assigned to do. Figure 2 shows some industrial tasks assigned to robots and the technology level required to complete the task.

You should view the drawings of Figure 1 to recognize the differences in appearance of these classifications of robots and try to visualize how they might perform the tasks listed in Figure 2. Also, from Figure 2, notice that high-technology robots can perform all the tasks which medium- and low-technology robots can perform.

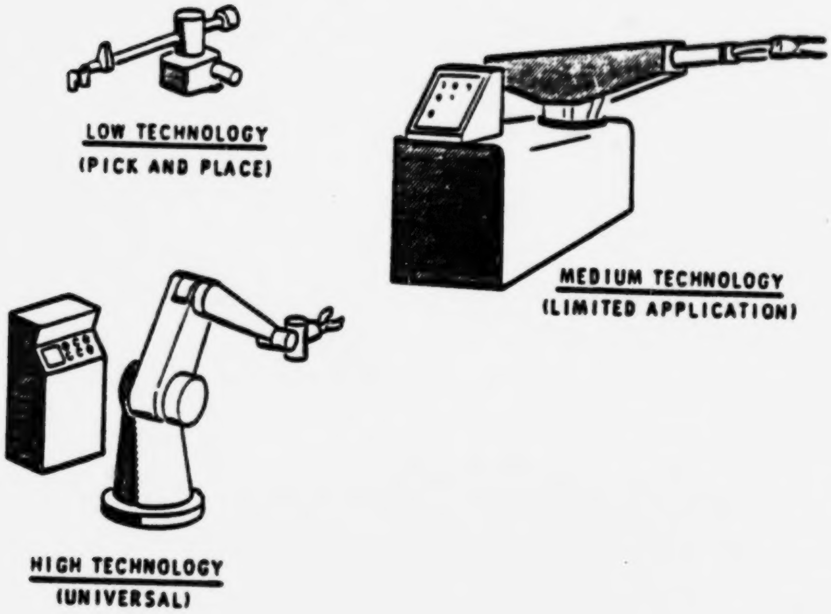

LOW TECHNOLOGY
(PICK AND PLACE)
MEDIUM TECHNOLOGY
(LIMITED APPLICATION)
HIGH TECHNOLOGY
(UNIVERSAL)

Figure 1
Types of Industrial Robots

	TECHNOLOGY LEVEL		
TASK ASSIGNED	Low	Medium	High
Material Handling	X	X	X
Press Operations	X	X	X
Injection Molding	X	X	X
Machine Load & Unload	X	X	X
Assembly (simple)	X	X	X
Die Casting		X	X
Spray Painting			X
Arc Welding			X
Spot Welding			X
Forging			X
Palletizing			X
Inspection (Body & Panels)			X
Complex Parts Assembly			X

Figure 2
Robot Classification By Task

LEARNING ACTIVITIES

After reading the previous Background Information and viewing the robot furnished in the High Technology Mobile Laboratory, answer the following:

1. What type of end-effector is on the robot? ______________________

   ____

2. In what technology-level category does this robot fit best, and why? ______________________________________

   ____

   ____

3. List at least eight "tasks" which a robot of this technology-level category might be able to perform.

   ____________________ ____________________

   ____________________ ____________________

   ____________________ ____________________

   ____________________ ____________________

   ____________________ ____________________

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 3

LAP TITLE: Introduction: Coordinate-Systems Categories

PREREQUISITE LAP (S) : 1 and 2

LAP PURPOSE: To provide definitions and background information regarding the various coordinate systems as used to describe a robot's movements.

COMPETENCIES TO BE DEVELOPED:

1. Define various basic terms associated with a robot's coordinate-system movements.
2. Identify various features and capabilities associated with the rectilinear, cylindrical, spherical and anthropomorphic robotics coordinate systems.

BACKGROUND INFORMATION

Definitions

Following is a list of terms and definitions which apply to robots' different coordinate systems.

Coordinate System - A method of defining the directions and dimensions of the space through which a robot's end-effector can move.

Rectilinear Coordinates - The movements are linear (straight line), one in each of the three planes X, Y, and Z.

Cylindrical Coordinates - The movement in two planes is linear (straight line) and in the third it is rotational.

Spherical Coordinates - The movement in one plane is linear (straight line), while in the other two planes it is rotational.

Anthropomorphic, or Revolute, Coordinates - All movements are rotational. Anthropomorphic means resembling human shape or characteristics. It describes the ability of a robot arm to move in a fashion similar to the human arm.

Degrees of Freedom - The total number of different movements, about axes, through which the robot can function. It is equal to the number of axes built into the robot.

Descriptions of Coordinate Movements

There are four dominant coordinate systems used in the designing of industrial robots. The first type is "rectilinear" and incorporates strictly linear X, Y and Z movements, which can be viewed in Figure 1-a. The second system is termed "cylindrical" due to it's revolving base but linear vertical and horizontal movements, as shown in Figure 1-b. The third system is based on "polar" or spherical coordinates. See Figure 2-a. The fourth, most human-like, is termed "revolute" or anthropomorphic. It consists of a revolving base, shoulder, elbow, and wrist movements, as shown in Figure 2-b.

In reviewing the four types on the basis of their motion, there are clearly three geometric shapes and an unlimited configuration. The rectilinear motion produces a square or cube, the cylindrical motion produces a cylinder, the polar a sphere, and the revolute can duplicate any of these and much more.

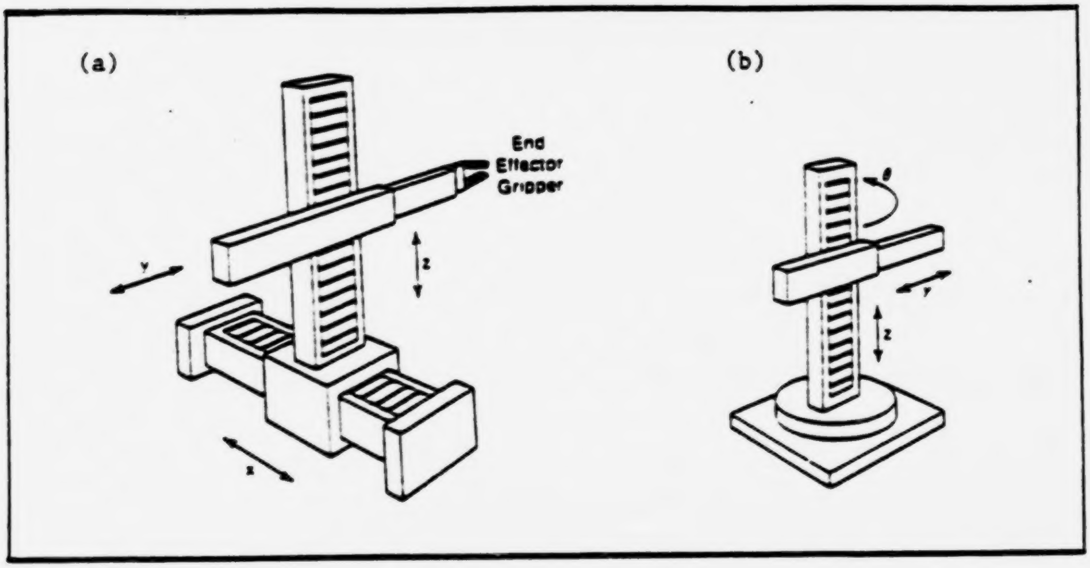

(a)
(b)
End Effector Gripper
y
z
x
θ
y
z

Figure 1. Coordinate Systems: (a) Rectilinear, (b) Cylindrical.
SOURCE: Richard C. Dorf, Robotics and Automated Manufacturing (Reston: Reston Publishing Co., 1983), pp. 52,55; figure 3-9, 3-14b.

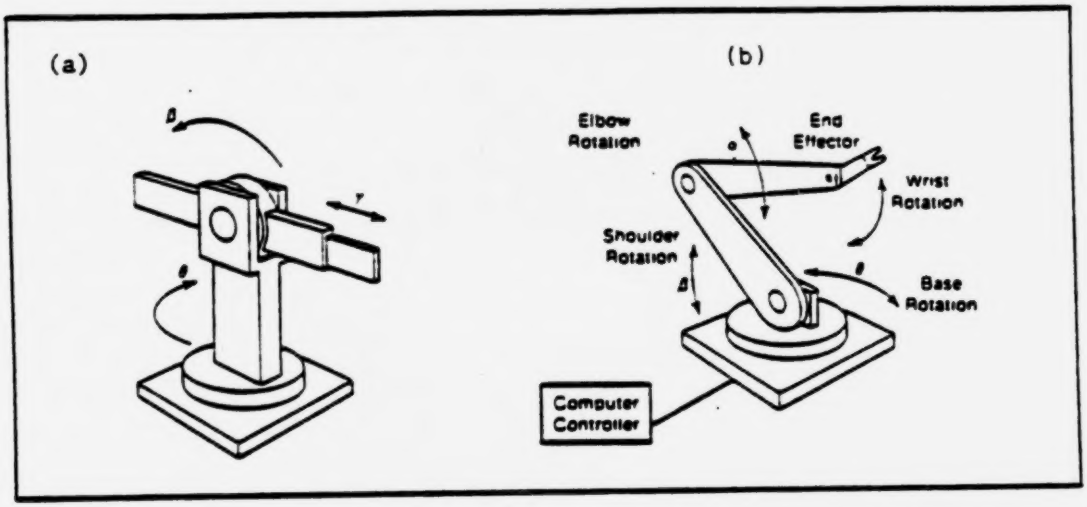

(a)
(b)
β
y
θ
Elbow Rotation
End Effector
Wrist Rotation
Shoulder Rotation
β
θ
Base Rotation
Computer Controller

Figure 2. Coordinate Systems: (a) Spherical, (b) Anthropomorphic.
SOURCE: Richard C. Dorf, Robotics and Automated Manufacturing (Reston: Reston Publishing Co., 1983), pp. 51,55; figures 3-8, 3-14a.

LEARNING ACTIVITIES

After reading the previous Background Information and viewing the robot furnished in the High Technology Mobile Laboratory, answer the following:

1. What coordinate system does this robot use, and how did you determine this? ____

   ____

   ____

2. How many "degrees of freedom" does this robot have? ____________

   ____

3. What geometric shapes can the motions of this robot produce?

   ____

   ____

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 4

LAP TITLE: The Teach Pendant: Connections, Power-Up and Simple Operations

PREREQUISITE LAP (S) : 1

LAP PURPOSE: To provide information regarding the proper system component interconnections, the powering of various parts and the basic operations involved in using the teach pendant to control the robot.

COMPETENCIES TO BE DEVELOPED:

1. Properly make all power and control-cable connections toward use of the teach pendant.
2. Identify significant "dos" and "don'ts" regarding operation/use of the robot.
3. Perform the basic operations/movements of the robot using the teach pendant.

BACKGROUND INFORMATION

Introduction

A teach pendant is a hand-held computer, usually attached to the robot controller, with which the operator can guide the robot through a series of points or in a motion pattern. Typically, a teach pendant can also be used to record the robot's moves for subsequent automatic actions by the robot.

A teach pendant is a circuit/device that permits a human to teach a machine. Some teach pendants have provisions whereby the program of moves stored in the teach pendant's memory can be "uploaded" to a host computer. Uploading is used when one wishes to save a previously teach-pendant developed program on disk. A program saved on disk can be used later to run the robot from the computer or, in some cases, can be "downloaded" into the teach pendant's memory for modification via editing.

Additional Information

Additional information on the proper connection of the robot system and on the teach pendant used with the robot furnished in the High Technology Mobile Laboratory is in Sections 1 through 6.2 on pages 1 through 10 of the XR SERIES TEACH PENDANT, Version 2.0 manual in Appendix A of this notebook. Read these sections now before beginning the Learning Activities of this LAP.

LEARNING ACTIVITIES

The following activities will guide you through a checkout of the connections of the robot, controller, and teach pendant. Then you will use the teach pendant to cause the robot to perform some basic moves. Remember that the objective is to BECOME FAMILIAR WITH THE TEACH PENDANT AND ROBOT. More complex work with the teach pendant and robot will be done in later LAPs.

When going through these activities it will be necessary to refer to the Teach Pendant manual in Appendix A of this notebook. References to this manual will be given by page number.

1. Proper connections of the teach pendant, controller, and host computer are shown in Figure 1a on page 3. Make these connections as illustrated. The host computer will not be used until a later LAP. However, you may connect it now, but do not turn on the computer.
2. The short jumper cable must be connected between the RS-232C Pendant and the RS-232C Controller jacks, as shown in Figure 1a on page 3. Make this connection.
3. The six flat ribbon cables coming from the robot's motors are labelled A, B, C, D, E, and F. These connect to the correspondingly labelled jacks across the lower center of the controller cabinet. Jacks G and H are used only when additional motors are used with other equipment around the robot. They will not be used in your work. Such motors could be controlled by pressing of the four lower right keys on the teach pendant.
4. Read the CAUTION paragraph on page 9. What does this mean in regard to the four lower right keys on the teach pendant? ________________

   ____
5. Plug the Controller power cord into a wall receptacle in the Laboratory. Turn the Controler POWER switch ON. The red light on the Controller will light. The small, round red light near the upper end of the teach pendant will indicate that it, too, is powered.
6. Turn the MOTORS switch of the Controller ON. If all has been done properly, the system is now ready for making robot segments moves by pressing appropriate keys.

   Note: If anything appears to go wrong during the following activities you may stop the robot by turning OFF either the MOTORS of the POWER switch on the Controller.

7. Now proceed to move the robot's segments by pressing either of the two direction keys for the gripper, wrist rotate, wrist flex, elbow flex, or waist rotate on the teach pendant.

   To cautions are important, which are:

   a) Do not rotate the waist beyond 90 degrees in either direction. Doing so may cause the motor cables to be damaged by the waist swivel plates.
   b) The space for the robot is limited. Therefore, do not allow the robot's arm or gripper to move against the wall!

8. Refer to page 10 of the manual. Then proceed to run the robot's motors at the most appropriate of the three available speeds in an effort to approach and grasp an object.
9. When you wish, you may press the SHIFT and GO HOME keys simultaneously! This will cause the robot to go to a fixed retracted position called "Hard Home".
10. Any of the steps 7, 8, and 9 may be repeated again for more practice.
11. When finished using the robot press the waist rotate move key to cause the robot's arm and hand assembly to be extended above the "+X" axis of the grid table. Then press the SHIFT and GO HOME keys simultaneously! This will cause the robot to go to a retracted position which is called "hard home." Lastly, turn power OFF in the following order:
    a) MOTORS switch
    b) MAIN POWER switch
    c) Unplug controller power cord.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 5

LAP TITLE: The Robot's Main Structure

PREREQUISITE LAP (S) : 4

LAP PURPOSE: To provide basic information regarding the structure, dimensions and features of the robot you will be using.

COMPETENCIES TO BE DEVELOPED:

1. Define various basic terms applicable to the most significant structural parts of the robot.
2. Identify and provide the dimensions and/or the functions of various parts of the robot.

BACKGROUND INFORMATION

Introduction

The major segments of the robot furnished in the High Technology Mobile Laboratory will be identified and briefly described in this LAP. Only the major, or most important, segments will be discussed. They are as follows:

1. Base -- The base is that part which is fastened to the table. It contains the waist axis of rotation and houses the waist drive motor "F".
2. Torso, or Shoulder Assembly -- The torso, or shoulder assembly, is that part which carries the bicep flex drive motor "E", the elbow flex drive motor "D", the wrist flex drive motor "C", and several sproket/chain drive linkages. It also contains the bicep flex axis of rotation.
3. Bicep, or Upper Arm -- The bicep constitutes part of the extension capability of the robot. Its length is from the bicep axis to the elbow axis.
4. Forearm -- The forearm consistutes another part of the extension capability of the robot. Its length is from the elbow axis to the wrist axis.
5. End-Effector -- The end-effector assembly is often referred to as the "business end" of the robot. It flexes about the wrist flex axis. Several different end-effectors are available for the robot you are using. However, the one now on the robot should consist of:
   a) the wrist rotate drive motor "B", with wrist rotate axis,
   b) the gripper drive motor "A", and
   c) a two-finger gripper.
6. Controller -- The robot controller contains controlling and motor drive circuitry for the robot. It forms the major link between the teach pendant and the robot. The control circuitry includes that which keeps track of the direction and number of revolutions of the motors by using the signals sent back to it from the encoder discs on the motors.
7. Encoder Discs -- There are six encoder discs on the robot, one attached to each of the six drive motors. The slots in these discs, along with the optical couplers over the discs' edges, enable the robot controller to keep track of both the distance and direction

of rotation of the motors.

8. Motors -- There is one motor for each of the six axes on the robot. Each motor drives a part of the robot through a speed reduction gearbox, whose speed reduction ratio is stamped on either the gearbox or the motor. The use of the gearboxes reduces the speed while increasing the torque which the motors can deliver to the robot segments. Because the encoder discs are attached to the motors, rather than to the output of the gearboxes, the controller can make smaller incremental measurements of distances moved by segments of the robot.
9. Limit Switches -- Most of the robot's driven segments (waist, upper arm, forearm, etc.) are equipped with limit switches. These switches "sense" when a segment has reached a limit of rotation or position. Several switches are also used to determine the "Hard Home" position of the robot.

Robot Actuators

An actuator is a motor or transducer that converts electrical, hydraulic, or pneumatic energy to effect movements of a robot's segments. Motors can be electrical, hydraulic, or pneumatic. They have rotary motion as their output. However, if translational (back and forth) movement is needed, the usual practice is to use hydraulic or pneumatic cylinders with pistons.

Each of the three types of actuators has its advantages and disadvantages. Some of these are listed below. *

A) Electric Actuators

Advantages

1) Electric actuators are fast and accurate.
2) It is possible to apply sophisticated control techniques.
3) Easily available and relatively inexpensive.
4) Simple to use.

Disadvantages

1) Require gear trains or the like for transmission of power.
2) Gear backlash limits precision.
3) Electric arcing might be a problem.
4) Power limit.

B) Hydraulic Actuators

Advantages

1) Large lift capacity

2) Moderate speed.
3) Oil is incompressible; hence, once positioned, the joints can be held motionless.
4) Offers accurate control.

Disadvantages

1) Hydraulic systems are expensive
2) They pollute the workspace with fluids and noise.
3) Not suitable for really high speed cycling.

C) Pneumatic Actuators

Advantages

1) Relatively inexpensive.
2) High speed.
3) They do not pollute the workspace with fluids.
4) Can be used in laboratory work.

Disadvantages

1) The compressibility of air limits their accuracy.
2) Noise pollution still exists.
3) Leakage of air is a major concern.
4) Additional air filtering system and drying system are needed.
5) Increased maintenance and construction requirements.

Additional Information

More information on the robot you are using may be found in Sections 19 and 42 of "Hands-On-Information to ROBOTICS, The Manual for the XR-Series-Robots.

---

* SOURCE: Edward Kafrissen and Mark Stephans, Industrial Robots and Robotics (Reston: Reston Publishing Co., 1984) pp. 90-91

LEARNING ACTIVITIES

The "Background Information" section of this LAP contains a total of 16 underlined items, each referring to a special major segment, or part, of the robot furnished in the High Technology Mobile Laboratory. The following page contains a crude sketch of the robot with 16 arrowed blanks. Your task is to read the "Background Information" section of this LAP and to place the appropriate underlined word or words in each blank of the sketch of Figure 1.

As an aid to better visualization of the segments or parts of the robot, you may wish to move some of its segments or parts. This can be done by following the connections, powering, and simple moves suggested in steps 1 through 7 of LAP #4.

When finished using the robot, complete the shut-down procedures of Step 11 in LAP #4.

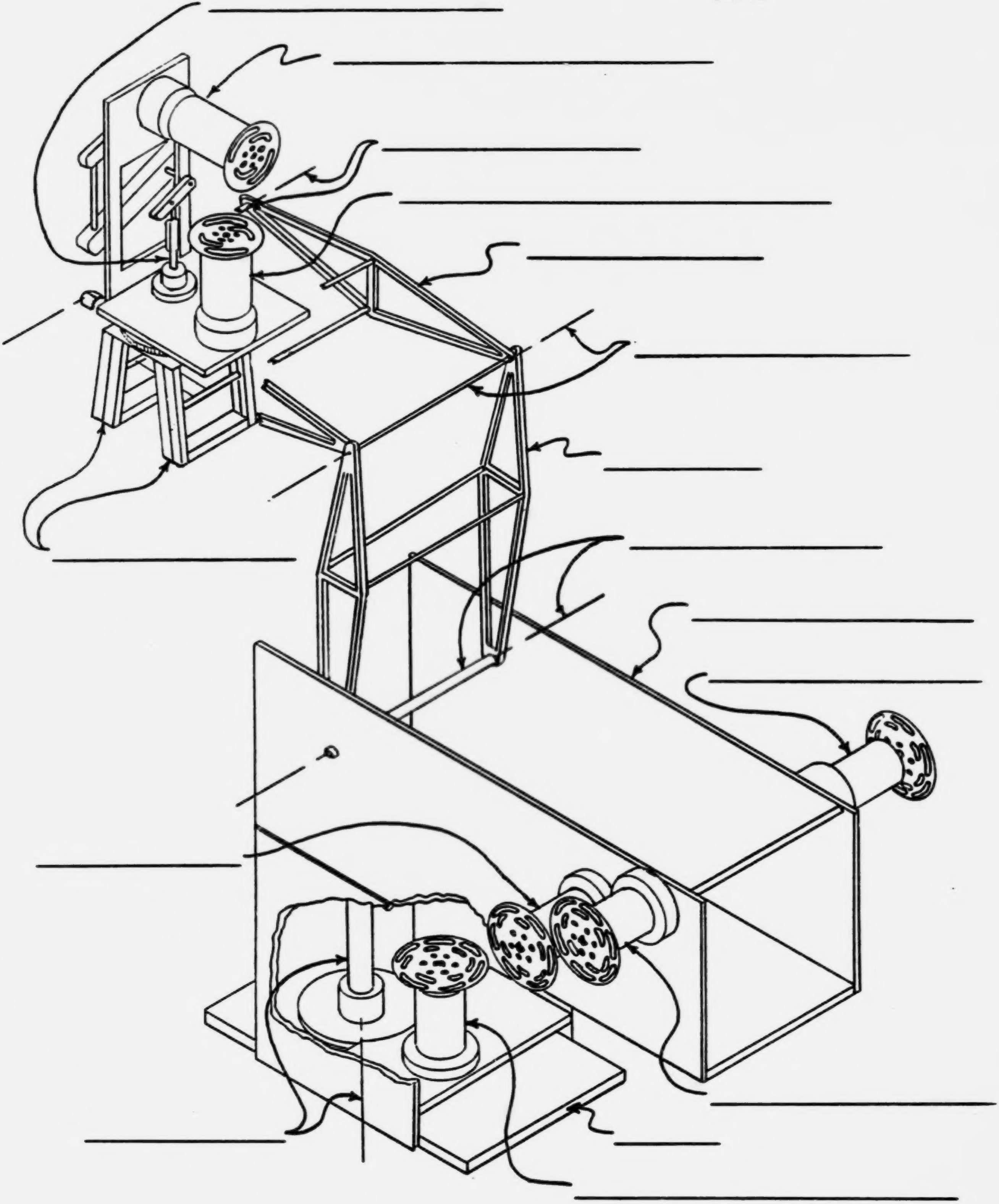


Figure 1. Sketch Showing Major Segments or Parts of Robot You Are Using.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 6

LAP TITLE: Robotics Maintenance And Troubleshooting

PREREQUISITE LAP (S) : 5

LAP PURPOSE: To provide information regarding typical practices in the area of troubleshooting and maintenance of robotic systems.

COMPETENCIES TO BE DEVELOPED:

1. Be able to identify potential problems and to take corrective actions toward keeping a robotic system in good working order.
2. Be able to perform simple maintenance work on the robotic system as directed by the instructor.

BACKGROUND INFORMATION

Introduction

Maintenance and troubleshooting of robotic systems, and any related machines which the robot may serve, is a very important part toward determining the performance and cost effectiveness of a robot. This LAP covers some of the concepts and considerations pertaining to maintenance and troubleshooting of robots. The following terms and comments do not cover all that may be involved, but should provide you with some understanding of the importance of this topic.

Terms and Definitions

1. Maintenance Schedule --- A recorded account of all servicing, repairs, and other maintenance performed on a robot. Such a record is important toward determining the cost of up-keep, the maintenance to be anticipated in the future, and any possible improvements regarding the robot or its use.
2. Payback Period --- The time, typically in years, that is required for the savings attained by using a robot to equal its original cost. This is a very simple definition because other costs such as interest, inflation, etc., are not taken into account in this definition.
3. Payload Handling Capabilities --- The payload of a robot basically concerns the mass or weight which the robot can grasp, lift, or handle safely. This can be effected by the reach, speed, acceleration/deceleration, etc., with which the robot must handle the load. These, in turn, can effect the reliability and life of the robot.
4. Sensors --- The type and number of sensors needed for a robotics installation depend upon the individual application. Sensors may be used to determine (sense) the presence/absence of parts which the robot is to handle, the position, direction, speed, etc., of parts and of the robot itself, and/or a wide variety of other conditions. Their calibration, adjustment, and maintenance can greatly effect the robot's performance.

Additional Comments on Maintenance

A wide variety of other maintenance and troubleshooting tasks may be involved in the up-keep of robots and related machines. Several of the possibilities are listed here.

1. Checking for proper positioning of limit switches and other sensors.
2. Checking for wear in joints and other linkages.
3. Adjustment of cable, belt, and/or chain drive tension.

4. Cleaning and lubrication of robot parts.
5. Checking of electrical cables and connections.
6. Checking of hydraulic or pneumatic hoses and connections.
7. Measuring/determining the accuracy and repeatability of end-effector positioning.
8. Checking the security of keys, set screws, and other fastens.

Additional Information

More information on maintenance and troubleshooting regarding the robot furnished in the High Technology Mobile Laboratory is contained in Sections 13, 14, 34, and 36 of "Hands-On-Introduction to ROBOTICS, The Manual for the XR-Series Robots".

LEARNING ACTIVITIES

Any maintenance or troubleshooting activities actually performed by a student for this LAP should be limited to those suggested by your instructor. A limited set of small tools is available for some of the possiblly needed adjustments.

The student is encouraged, however, to make a thorough inspection of the robot in an effort to identify any maintenance that may be needed. In addition, the student might wish to determine the whereabouts and function(s) of the limit switches and other sensors on the robot.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 7

LAP TITLE: Teach Pendant: Robot Waist, Bicep and Elbow Actions

PREREQUISITE LAP (S) : 4 and 5

LAP PURPOSE: To provide information on and experience in operation of the waist, bicep and elbow axes segments of the robot and thereby determine the range of these movements.

COMPETENCIES TO BE DEVELOPED:

1. Define basic and/or analogous terms associated with various axes of a robot.
2. Determine the range of the waist, bicep and elbow movements of the robot using the teach pendant.

BACKGROUND INFORMATION 7.1.1

Introduction

The robot furnished in the High Technology Mobile Laboratory has six degrees of freedom. For an anthropomorphic (human-like arm) type of robot this means there are six axes of rotation, or six rotational parts. The three of concern in this LAP form a group called the spatial axes, or segments, because they largely determine the total space throughout which the end-effector can move.

The three spatial axes or segments, the analogous terms used in robotics to identify them, and a brief description of their functions are as follows:

1. Waist Rotate --- also called Base: The axis of rotation is vertical. The entire arm and end-effector assembly rotates side-to-side about a bearing surface in the base of the robot. This action is much like rotation of the human upper body about the waist.
2. Bicep Flex --- also called Shoulder: The axis of rotation is horizontal, but this axis moves side-to-side with waist rotate. Thus, the bicep flex provides much of the vertical movement of the entire arm and end-effector assemply. The human bicep is the upper arm, from shoulder to elbow.
3. Elbow Flex --- also called, simply, Elbow: The axes of rotation is horizontal, but this axis moves side-to-side with waist rotate. The elbow joint (axis) carries the forearm, much like that of the human. The forearm, from elbow to wrist, adds both extension and versatility of applications to the robot.

The above descriptions should help the reader visualize why these three axes or segments are called the spatial axes. These "revolute" axes increase the potential tasks which the robot can perform.

Additional Information

The STALL Routine may be encountered in activities of this LAP. Therefore, read Section 10 on page 27 of the XR SERIES TEACH PENDANT, Version 2.0 manual in Appendix A of this notebook.

LEARNING ACTIVITIES

The following activities will help you determine the range of movement of the spatial axes and thereby understand how they add versitility to the robot. The "reach" of a robot is defined as the maximum extension which can be achieved by its arm segments. This sometimes excludes, but sometimes includes, the extension due to any segment(s) beyond the wrist joint. This is because most robots can support a variety of end-effectors, most of which may be detached at the wrist.

1. Prepare the robot, controller, and teach pendant for operation by making connections and applying power as described in steps 1 through 6 of LAP #4. The proper connections are shown in Figure 1a on page 3 of the XR SERIES TEACH PENDANT, Version 2.0 contained in Appendix A of this notebook.
2. Press the proper move keys on the teach pendant to cause the robot's upper arm and forearm to extend straight vertically. Also use the wrist flex key to cause the "hand" to be vertical, i.e., straight in line with the two segments of the arm.

Note: In the following procedures any one of the arm or wrist segments may reach its limit of rotation and cause movements to stop. When this happens the system will go into the STALL Routine. The word StALL will appear on the five right segments of the teach pendant display. To recover from this routine, you may, in this LAPs non-entered use of the teach pendant, simply press the RESET button on the teach pendant. This will cause the system to go through a dELAy during which corrective movements take place. Then a P, indicating the Play mode, appears in the left segment of the display which indicates that you can again use any of the move keys.

3. Now use the bicep, elbow, and wrist keys as needed to determine the range of angular movement of the bicep joint as indicated on the protractor attached at the bicep axis. The elbow and wrist flex may need to be moved during this to avoid the STALL Routine! The bicep moved through a total of ______ degrees.
4. Return the wrist and both arm segments to the straight vertical position described in step 2 above.
5. Now use the elbow, bicep, and wrist keys as needed to determine the range of angular movement of the elbow joint as indicated on the protractor attached at the elbow axis. The bicep and wrist flex may need to be moved during this to avoid the STALL Routine! The elbow moved through a total of ______ degrees.

6. Because of the restrictions of room in the Laboratory and the cautions cited in step 7 of LAP #4, the range of waist angular movement will not be measured!
7. Using a measuring tape, measure the length of the upper arm (from bicep axis to elbow axis). The upper arm length is ____.
8. Measure the length of the forearm (from elbow axis to wrist axis). The forearm length is ______.
9. Return the wrist and both arm segments to the straight vertical position described in step 2 above.
10. Measure the total arm length (from bicep axis to wrist axis). This arm length (reach) is ______. Does this equal the sum of lengths recorded in steps 7 and 8 above? ______. If not, explain any difference!
11. Measure the length of arm plus hand assembly (from bicep axis to end-effector tip). This arm plus hand assembly length (reach) is ______.
12. When finished using the robot rotate the waist so the arm is extended above the "+X" axis of the grid table, then press SHIFT and GO HOME simultaneously!, allow the robot to go home, and then turn power OFF in the following order.

    a) MOTORS switch.

    b) POWER switch.

    c) Unplug Controller power cord.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 8

LAP TITLE: Teach Pendant: Robot Wrist Flex and Rotate Actions

PREREQUISITE LAP (S) : 4 and 5

LAP PURPOSE: To provide information on and experience in operation of the flex and rotate wrist joints of the robot and thereby determine the range of these movements.

COMPETENCIES TO BE DEVELOPED:

1. Define basic and/or analogous terms associated with the wrist axes of a robot.
2. Measure/determine the range of the flex and rotate movements of the robot through use of the teach pendant.

BACKGROUND INFORMATION

Introduction

The robot furnished for your use has six degrees of freedom. This means there are six joints or moving segments on the robot. In LAP #7 you were introduced to the spatial axes, which consist of the waist flex, the bicep flex, and the elbow flex movements.

This LAP will concern two of the remaining three segments on the Laboratory's robot. These two segments, analogous terms used in robotics to identify them, and a brief description of their functions are as follows:

1. Wrist Flex -- also called Wrist Pitch: The axis of rotation is horizontal. It is located at the far end of the forearm, which is sometimes referred to as the wrist joint. Rotation about this axis forms up and down (vertical) movement of the end-effector, or "hand" assembly.
2. Wrist Rotate -- also called Wrist Roll: The axis of rotation typically points through, or is aligned with, the center of the end-effector. The position and direction of this axis is similiar to the straightened out center finger of the human hand. Rotation about this axis enables "rolling" of the end-effector or "hand" assembly. This is often needed for parts placement by the robot.

One additional wrist axis is available on or for some robots. Where needed, the wrist may also have Wrist Yaw. This is most easily described as side-to-side swing of the end-effector or "hand" assembly. This axis adds great versitility to the robot. It permits the end-effector to reach around objects or into corners, such as may be necessary in arc welding applications.

Additional Information

More information on the wrist flex and wrist rotate segments may be found in Section 22 of "Hands-On-Introduction to ROBOTICS, The Manual for the XR-Series Robots." Also, because the STALL Routine may be encountered in the activities of this LAP, the student should read Section 10 on page 27 of the XR SERIES TEACH PENDANT, Version 2.0 manual in Appendix A of this notebook.

LEARNING ACTIVITIES

The activities of this LAP will help to familiarize you with the wrist flex and wrist rotate operations, movements, and limitations of the robot you are using. It is assumed that you understand how the proper connections and powering of the robot, controller, and teach pendant are made, as described in LAP #4.

1. Prepare the robot system for use by following steps 1 through 6 of LAP #4.
2. Press the proper move keys on the teach pendant to cause the robot's upper arm and forearm to extend and the waist to rotate until the "hand" assembly is in a good position for viewing. It is probably best to have the forearm and "hand" assembly in the vertical position to permit easy observation of the "hand" assembly.

Note: In the following procedures, if any segment of the robot reaches its limit of rotation the STALL Routine will be encountered. If that happens, reread the "note" in step 2 of LAP #7 on how to recover from the STALL Routine.

3. Now use the teach pendant's wrist flex keys (one for each direction) to cause the "hand" assembly to flex fully in each direction, and eastimate the range of angular roation through which the wrist can flex. This total angular rotation of wrist is approximately ______degrees.
4. Return the "hand" assembly to the position of step 2 above.
5. Now use the teach pendant's wrist rotate keys (one for each direction) to cause the wrist rotate action, i.e., rotation of the end-effector. After doing so, estimate the range of angular rotation of the end-effector. The total angular rotation of the end-effector is approximately ______degrees.
6. Return the end-effector to the approxmiate center of its rotational range.
7. When finished using the robot, complete the shut-down procedures of step 11 of LAP #4.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 9

LAP TITLE: Robotic End-Effectors and Actions

PREREQUISITE LAP (S) : 4 and 5

LAP PURPOSE: To provide information regarding the various end-effectors useable on robots and experience in using the robot's end-effector(s).

COMPETENCIES TO BE DEVELOPED:

1. Define types and features and potential uses of various robotics end-effectors.
2. Determine various parameters, e.g., width, opening, force, etc., of end-effector(s) on the robot which you are using.

BACKGROUND INFORMATION

Introduction

As stated in LAP #5, the end-effector is the "business end" of a robot. An end-effector may be defined as any actuator, gripper, or mechanical device which is attached at the wrist joint of a robot for the purpose of picking up, moving, or positioning an object. In this case the object, which constitutes the payload, could be any of many things such as a workpiece to be machined, a part to be assembled, or the wire which comes out of the nozzle of a wire-feed welder. The end-effector is what "handles" that object.

The design and manufacture of end-effectors is a somewhat specialized area within the field of robotics. This is because of the many applications wherein robots perform a wide variety of tasks. For example, a robot used to insert and remove parts from a lathe may require a differently shaped end-effector for each type of part machined on that lathe. Thus, the end-effectors must be easily interchangeable for the different parts production runs done on that lathe.

The design of end-effectors is as varied as the robot applications themselves. A few broad categories of end-effectors are shown in Figure 1. Notice that they may be pneumatically (vacuum), electrically, or mechanically driven. Also, they may be a power tool, measuring device, or some sort of gripper. The gripping-type end-effector is the most often used because it can be made to be able to grasp a wide variety of shapes and sizes of objects. See Figure 2.

Grippers can usually be classified as two-finger or three-finger, and each can be made to grasp an object internally (hollow object) or externally. The three-finger gripper is often used where the object is circular or spherical and where the object must be centered between the fingers.

Gripper-Type End-Effector Measurands

Generally, there are three things which can be measured on a gripper. They are:

1. Jaw Size or Finger Width -- This refers to the diameter of an internal gripper when closed, or to the contact distance along the width of a finger.

2. Opening or Span -- This opening is the distance between fingers when fully opened. Span refers to both the minimum and maximum opening, and must be given by stating two values, e.g., 1/8" to 1-1/4".
3. Pinch Force -- This is the force which the gripper fingers can apply to the object being grasped.

Additional Information

More information on "hand" assemblies of the robot you are using may be found in Sections 20 and 22 of "Hands-On-Introduction to ROBOTICS, The Manual for the XR-Series Robots." Also, because the STALL Routine may be encountered in the activities of this LAP, the student should read Section 10 on page 27 of the XR SERIES TEACH PENDANT, Version 2.0 manual in Appendix A of this notebook.

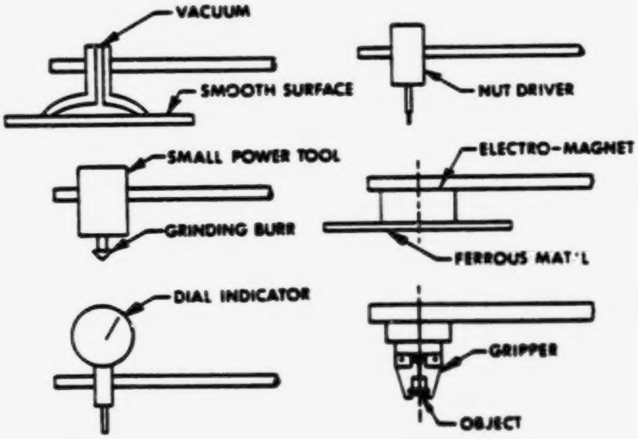

VACUUM
SMOOTH SURFACE
NUT DRIVER
SMALL POWER TOOL
GRINDING BURR
ELECTRO-MAGNET
FERROUS MAT'L
DIAL INDICATOR
GRIPPER
OBJECT

Figure 1. End-effectors. Courtesy of Mack Corp.
SOURCE: Edward Kafrissen and Mark Stephans, Industrial Robots and Robotics (Reston: Reston Publishing Co., 1984), p. 156, figure 6.3.

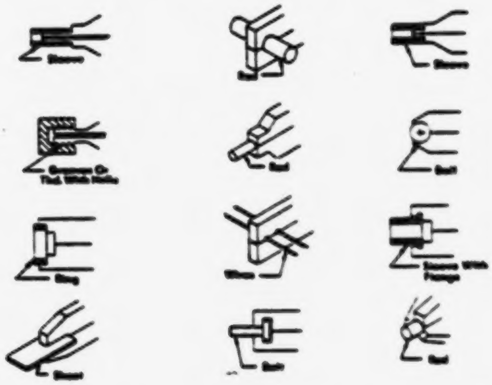


Figure 2. The B.A.S.E. Robotic System Showing Grippers. Courtesy of Mack Corp.
SOURCE: Edward Kafrissen and Mark Stephans, Industrial Robots and Robotics (Reston: Reston Publishing Co., 1984) p. 157 figure 6.5

LEARNING ACTIVITIES

The activities of this LAP will enable you to determine some of the parameters of the end-effector which is on the robot furnished in the High Technology Mobile Laboratory. It is assumed that you understand how the proper connections and powering of the robot, controller, and teach pendant are made, as described in LAP #4. Also, it is assumed that the two-finger gripper is attached as the robot's end-effector.

1. Prepare the robot system for use by folowing steps 1 through 6 of LAP #4.
2. Press the proper move keys on the teach pendant to cause the robot's upper arm and forearm to extend and the waist to rotate until the "hand" assembly is in a good position for viewing. It is probably best to have the "hand" assembly horizontal so you can easily see and reach the gripper.

Note: If, in any of these procedures, any of the robot's segments reaches its limit of rotation the STALL Routine will be encountered. If that happens, reread the "note" in step 2 of LAP #7 on how to recover from the STALL Routine.

3. Now, using the teach pendant's "grip" keys (one to open and one to close the gripper), cause:
    a) The gripper to close as far as it will; and then measure and record the minimum opening between the fingers. This measured minimum opening is ________inch.
    b) The gripper to open as far as it will, and then measure and record the maximum opening between the fingers. This measured maximum opening is ________inch(es).
4. Use the measured mimimum and maximum opening of the gripper fingers to state the span of gripper opening. This span is ____________ to ____________.
5. Having the gripper fingers open, measure and record the finger width. The finger width is ________inch(es).
6. The gripper pinch force of the robot furnished in the Laboratory is not easily measured. Also, the pinch force of such a gripper is not a constant value. The levers which open and close this gripper cause the pinch force to depend upon how far the fingers are open or

closed. This pinch force will not be measured!

7. When finished using the robot, complete the shut-down procedures of step 11 of LAP #4.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 10

LAP TITLE: The Robot's Working Envelope

PREREQUISITE LAP (S) : 4 and 5

LAP PURPOSE: To provide information regarding various robots' working envelope shape.

COMPETENCIES TO BE DEVELOPED:

1. Define the dimensional planes involved in the working envelopes or robots in general.
2. Determine and sketch the size/shape of the working envelope being used, through use of the teach pendant.

BACKGROUND INFORMATION

Introduction

The working envelope of a robot is the total three-dimensional space within which the end-effector of the robot can be positioned. Each of the four coordinate-system catergories (see LAP #3) produces a differently shaped working envelope. These four shapes are as follows:

1. The rectilinear coordinate system robot produces a rectangular box (X,Y, and Z axes) envelope.
2. The cylindrical coordinate system robot produces a cylinder-shaped envelope.
3. The spherical coordinate system robot produces a partial-sphere-shaped envelope.
4. The revolute coordinate system (anthropomorphic) robot also produces a partial-sphere-shaped envelope, but it is far more intricate than that of the spherical coordinate system robot.

Working Envelopes

Diagrams of the working envelopes of the spherical and the revolute (anthropomorphic) coordinate system robots are shown in Figures 1 and 2, respectively. Each of these figures shows both a plan (top) view and an elevation (front) view. The common terms from these two figures are:

1. Swing -- The angle through which the waist will rotate.
2. Horizontal Reach -- The distance, in the horizontal plane, from the bicep or shoulder axis to the end of the end-effector when all segments are fully extended.
3. Vertical Stroke -- In the spherical robot this is the vertical plane (up-and-down) angle through which the arm can rotate. However, in the anthropomorphic robot this is the vertical plane (up-and-down) distance through which the end-effector can move. In this case, because of the base height, the vertical stroke exceeds the horizontal reach.

Viewing Figures 1 and 2, we see two additional terms which appear only in Figure 1. These two terms are:

1. Horizontal Stroke -- The in-and-out distance through which the arm can move the end-effector. As can be seen in Figure 1, this movement is caused by extension and retraction of a linear actuator, such as a hydraulic cylinder-piston.

2. Vertical Reach -- The up-and-down distance through which the arm can move the end-effector. From Figure 1 it can be seen that this distance is effected by horizontal stroke. Thus, the vertical reach is expressed as the maximum up-and-down angle.

The above discussion of the working envelopes shown in Figures 1 and 2 illustrates an important characteristic of the anthropomorphic robot. This robot can produce far more detailed movements within its working envelope than can the spherical coordinate system robot. Also, it can work closer to its own base.

Robot Work Cell

The work cell of a robot involves the area in which the robot performs tasks on or with other machines within a factory. The related machines can include a wide variety of possibilities, such as shown in Figure 3. In this illustration we see that the robot must be able to reach most of the elements surrounding it. Figure 4 shows how robots can perform tasks along an assembly line. Here the robots might also reach to the rear of their bases to pick up parts for assembly.

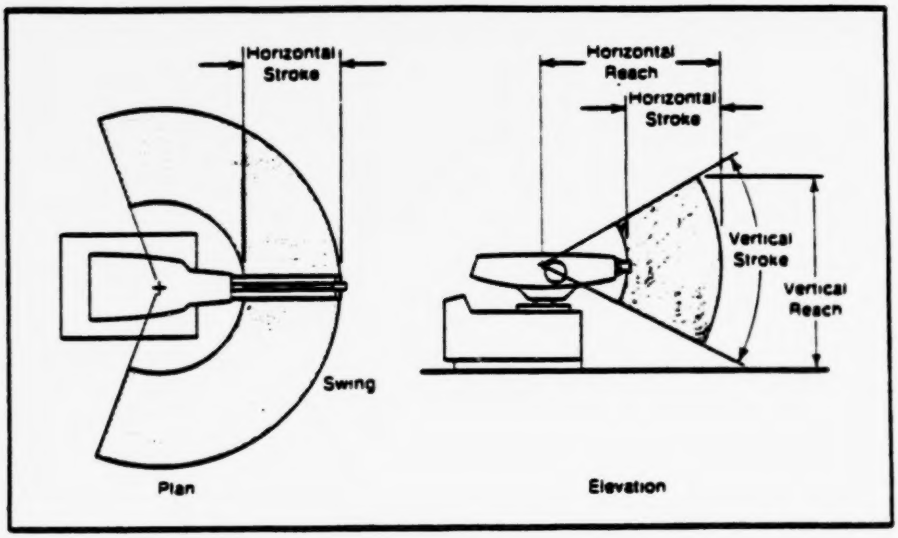

Horizontal Stroke
Horizontal Reach
Horizontal Stroke
Vertical Stroke
Vertical Reach
Swing
Plan
Elevation

Figure 1. Spherical Robot Work Envelope

SOURCE: Richard C. Dorf, Robotics and Automated Manufacturing (Reston: Reston Publishing Co., 1983), p. 58, figure 3-17.

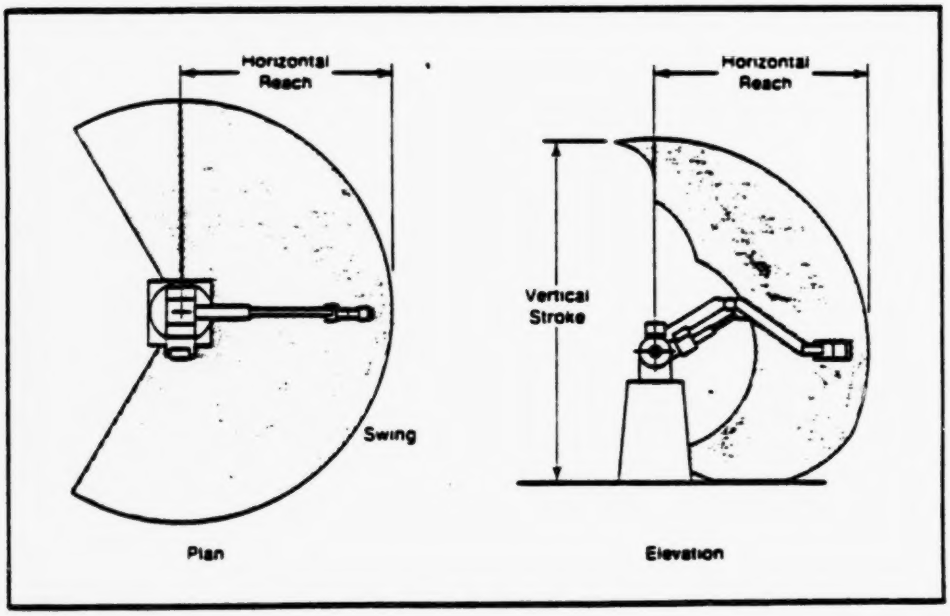

Horizontal Reach
Horizontal Reach
Vertical Stroke
Swing
Plan
Elevation

Figure 2. Anthropomorphic Robot Work Envelope

SOURCE: Richard C. Dorf, Robotics and Automated Manufacturing (Reston: Reston Publishing Co., 1983), p. 59, figure 3-18.

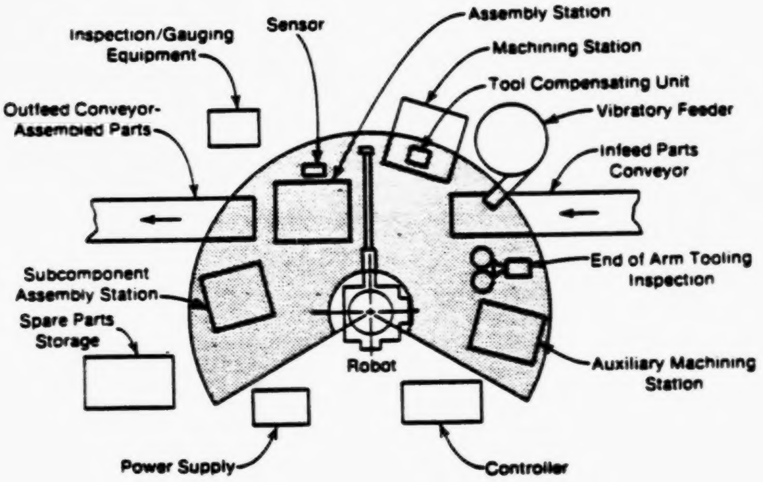

Inspection/Gauging Equipment
Sensor
Assembly Station
Machining Station
Tool Compensating Unit
Vibratory Feeder
Outfeed Conveyor-Assembled Parts
Infeed Parts Conveyor
End of Arm Tooling Inspection
Subcomponent Assembly Station
Spare Parts Storage
Robot
Auxiliary Machining Station
Power Supply
Controller

Figure 3. The Potential Elements of a Robot System

SOURCE: Richard C. Dorf, Robotics and Automated Manufacturing (Reston: Reston Publishing Co., 1983), P. 59, figure 3-19.

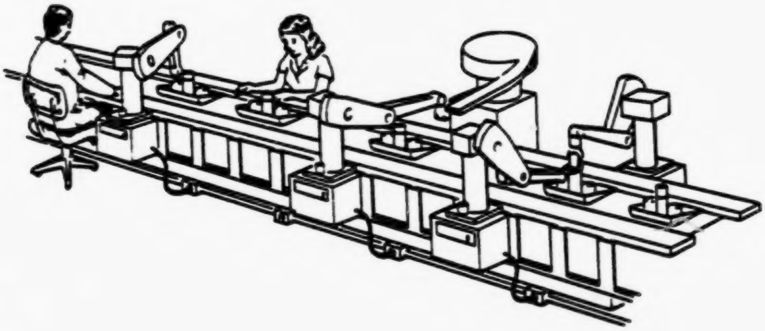


Figure 4. Robot Arms with Revolute Coordinates May be Used Alongside Humans on an Assembly Line

SOURCE: Richard C. Dorf, Robotics and Automated Manufacturing (Reston: Reston Publishing Co., 1983), p. 60, figure 3-20.

LEARNING ACTIVITIES

The activities of this LAP will involve your determining some of the dimensions of the working envelope of the robot furnished in the High Technology Mobile Laboratory. It is assumed that you understand how the proper connections and powering of the robot, controller, and teach pendant are made, as described in LAP #4. Also, it is assumed that the two-finger gripper is attached as the robot's end-effector.

1. Prepare the robot for use by following steps 1 through 6 of LAP #4.
2. Press the proper move keys on the teach pendant to cause the robot's upper arm, forearm, and "hand" assembly to extend in a straight horizontal position. The waist may be rotated so the entire arm extends outward along the "+X" axis of the grid table. The gripper may remain either open or closed throughout the activities of this LAP.

Note: If any of the robot's segments reaches its limit of rotation in this or following procedures the STALL Routine will be encountered. If that happens, reread the "note" in step 2 of LAP #7 on how to recover from the STALL Routine.

3. With the entire arm extended as described in step 2 above, measure and record the horizontal reach (see Figure 2) of the robot. This measured maximum horizontal reach, which may be labeled radius R, is _______inches.
4. The minimum radius, which may be labeled r, of this robot is the minimum horizontal distance from the bicep axis to the gripper fingers' tips. Proceed to maneuver the bicep flex, elbow flex, and wrist flex segments of the robot to the position where this distance can be measured. Then measure the distance from the bicep axis to the gripper fingers' tips. This measured minimum horizontal reach r is _______inches.
5. Re-extend the entire arm and "hand" assembly to the position described in step 2 above.
6. Now move the entire arm and "hand" assembly straight vertically. Then measure and record the vertical stroke (see Figure 2) of the robot. This measured vertical stroke is _______inches.

7. Optional: You may wish to move the upper arm, forearm, and wrist flex segments backward over the robot's base. These movements would illustrate the ramining possible dimensions of the robot's working envelope, as shown in the elevation view in Figure 2.
8. The robot you are using is capable of a "waist rotate" movement of 360 degrees. However, due to space restrications (the wall) in the Laboratory and possible danger of pinching motor cables, this movement should not be made.
9. Using the "waist rotate" capability of 360 degrees and the measurements made in steps 3, 4, and 6 above (and optional step 7 if done) sketch, in the space below, both the plan and elevation views of the working envelope for this robot. Place the proper dimensions on the views of this sketch.

Plan View

Elevation View

10. When finished using the robot, complete the shut-down procedures of step 11 of LAP #4.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 11

LAP TITLE: The "Hard" and "Soft" Home Positions/Movements

PREREQUISITE LAP (S) : 4

LAP PURPOSE: To provide information on the meaning of "home position" and experience in robot operations involving both the "hard" and a chosen "soft" home.

COMPETENCIES TO BE DEVELOPED:

1. Define various terms associated with "home position".
2. Perform operations involving both the "hard" home and in establishing a chosen "soft" home using the robot with teach pendant.

BACKGROUND INFORMATION

Introduction

The "home" position of a robot may be any of a variety of fixed or selected positions of the various segments or parts of the robot. A "home" postion can be used for a variety of purposes, including:

1. Establishing the safest position of each robot segment in case of power failure. In this case it may be referred to as the fail-safe home position.
2. Establishing an initializing, or calibrating, position for the robot. For this type of home the position of microswitches which are actuated by cams on various driven segments (driven sprokets for example) serve as reference points for the various segments. In some cases this is called HARD HOME, meaning that his home's position data is fixed by hardware.
3. Establishing a best position within the total cycle of the movements through which the robot is to move in performing its task(s). This type of home is usually called SOFT HOME (software home) because it is established in memory by the user.

Home Positions

As stated above, hard home is a fixed position due to its being determined by hardware. However, the controller must have some means of knowing when hard home is reached so that it knows when to stop movements. The technique by which this is known involves signals being sent to the controller from limit switches attached at predetermined locations on the robot. These signals "tell" the controller that a certain position has been reached.

The location of a soft home position is left to the judgement of the programmer. However, an often-used principle is to have the soft home location a short distance directly above a parts pick-up location within the robot's total cycle. Another possible location for soft home could be at the end of a particular move where some time delay is needed. In this case the robot would wait at this delay position until a signal is received to continue and to repeat its cycle.

Additional Information

More information on the hard home position and on how to establish a soft home position is contained in Sections 5.1, 7.1, and 7.2 in the

XR SERIES TEACH PENDANT, Version 2.0 manual contained in Appendix A of this notebook. The student is expected to read these sections before proceeding to the activities of this LAP. Also, if you are interested in how the robot's limit switches effect hard home, reading Section 14 of "Hands-On-Introduction to ROBOTICS, The Manual for the XR-Series Robots" may be helpful.

LEARNING ACTIVITIES

The activities of this LAP are meant to give the student a better understanding of the Hard Home position and some familiarity with procedures of setting a Soft Home position for the robot furnished in the High Technology Mobile Laboratory. If any segment of the robot should reach its limit of rotation in these activities, the STALL Routine will be encountered. If that happens, reread the "note" of step 2 in LAP #7 on how to recover from the STALL Routine.

1. Prepare the robot system for use by following steps 1 through 3 of LAP #4.
2. Perform and notice the reactions in the following procedures:
   a) Turn the MAIN POWER switch on the controller ON. The word dELAy will appear for a short time; the word SEt will flash for a very short time; and a P will then show on the left, in that sequence, on the teach pendant display. The teach pendant is now in the PLAY mode.
   b) Turn the MOTORS switch on the controller ON. The fingers of the gripper will open, if they are not already open.
   c) If you did not notice all of the reactions mentioned in a and b above, turn both switches OFF and then repeat those steps.

A. Hard Home Position Demonstration

3. Press the proper move keys on the teach pendant to cause the upper arm, forearm, and hand assembly to move to any extended position and the waist rotate key to rotate the waist some small angle (less than 90 degrees). This waist rotate should not be toward the laboratory wall because of space restrictions.
4. After the robot has been moved as suggested in step 3 above, pay particular attention to the location of some limit switches (the waist, bicep, and elbow driven sprokets are good examples) and then simultaneously press SHIFT and GO HOME. The robot should move to a retracted position and stop due to these switches being actuated.
5. If you did not succeed in seeing actuation of the limit switches while doing step 4 above, repeat step 3 and then step 4 again.
6. Be certain that the last key pressed on the teach pendant was GO HOME and that the robot has gone to Hard Home. This is necessary prior to establishing a Soft Home. When GO HOME was pressed, you should have seen the word HArd on the display.

B. Soft Home Demonstration

Setting a soft home requires a definite sequence of steps on the teach pendant. Follow these instructions carefully!

1. If the MAIN POWER and MOTORS switches of the controller are not ON, turn them ON.
   If these switches are already ON, press the RESET button.
   In either case, you should observe the display words and gripper finger movements described in step A-2-a and A-2-b above. The teach pendant should end up in the PLAY mode.
2. You must do a Hard Home before setting a Soft Home! Therefore, press SHIFT and GO HOME simultaneously. If the robot is already at hard home or just after it goes to hard home, you should see the word HArd flash on the display.
3. Press several chosen move keys on the teach pendant to move various segments of the robot to a desired Soft Home position. A position several inches above and in from the near edge of the grid table is suggested.
4. Press the SET HOME key on the teach pendant. This records the Soft Home position in memory.
   Note: If you forgot to do a hard home before pressing SET HOME the teach pendant will remind you by flashing the words dO HArd on the display.
5. Once the soft home is set, as in step 4 above, you can immedidately check it by pressing GO HOME. If the robot does not move, it is already at the soft home.
6. Press the LEARN key. L should appear at the left of the display. This indicates that the teach pendant is ready to learn some moves.
7. Now move several segments of the robot sequentially. After each move, press the ENTER key to record the moves in memory. Notice that the number of entries will be shown on the right of the display. You may wish to slow the robot by pressing the 10 or 1 key while making these moves.
8. When your chosen sequence of moves have all been entered, press the END key. This indicates the end of your program, and should cause the robot to go to the soft home of step 4 above.
9. The robot should now be ready to execute the program just recorded. Press the RUN key. The robot should move through the sequence of entered moves, pausing only momentarily at the Soft Home set in step

4 above.

10. To stop the robot, you can press the END key at any time during its cycle. It should cause it to stop at the soft home position. Once stopped, the sequence of moves can be started again by pressing the RUN key. Stop and restart the robot several times before proceeding.

C. Shut-Down

1. When finished with section B above, stop the robot by pressing the END key. Then press the RESET button on the teach pendant. This will remove all of the learned moves and the soft home from memory.
2. When finished using the robot, complete the shut-down procedures of step 11 of LAP #4.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 12

LAP TITLE: Teach Pendant: The "LEARN" A Sequence Mode

PREREQUISITE LAP (S) : 4 and 11

LAP PURPOSE: To provide information on and experience in using the teach pendant's LEARN mode toward establishing a chosen sequence of single axis movements of the robot.

COMPETENCIES TO BE DEVELOPED:

1. Describe the proper sequence of teach pendant commands needed to establish a sequence of LEARNed movements for the robot.
2. Choose/sketch/dimension a sequential movement pattern for the robot and input the proper teach pendant commands to cause that movements pattern to be executed by the robot.

BACKGROUND INFORMATION

Note: The learning activities of this LAP may be used in conjunction with those of LAP #13.

Introduction

The LEARN mode of a teach pendant is probably its most important feature. It enables an operator or programmer to record in memory the entire sequence of moves of all segments of the robot. The pressing of the move keys on the teach pendant merely causes the moves to take place so long as a key is pressed. However, this could be accomplished using an equal number of momentary-contact push-button switches.

As you may have noticed during the activities of previous LAPs, each time a move key is pressed a motor letter designation (A, B, C, D, E or F) appears on the display. Numbers also appear and will change as the key is pressed. Also, under proper conditions a "-" sign will preceed the number. These three items enable the teach pendant to record in memory the various moves which are to be LEARNed, but only when the teach pendant is in the LEARN mode.

Display Components

The significance of the above-mentioned three items is as follows:

1. The displayed letter indicates which motor is involved in a particular move. Though pressing two or more move keys at once will not cause simultaneous movement, pressing first one and then another and then ENTERing them together will cause those motors to RUN simultaneously. This will be covered more thoroughly in a later LAP.
2. The displayed number indicates the number of steps which the motor has moved or is away from a reference point, such as a home position.
3. The polarity sign indicates the direction which the motor has moved or is away from the reference point. Keep in mind, however, that a "+" is understood if no polarity sign is displayed, but a "-" sign will be displayed when appropriate. By way of example, if the waist rotate (motor F) on the robot furnished in the High Technology Mobile Laboratory is moved clockwise (as viewed from above the robot) far enough, a "-" sign will appear. This "-" and clockwise rotation agrees with that used in mathematics.

Thus, we see that we can "teach" the robot specific moves through use of the LEARN mode of the teach pendant. To "teach" a robot means to program it by guiding it through a series of moves and recording those moves in memory. Once in memory, the information can be used for future automatic action by the robot.

Additional Information

More information regarding the content being emphasized in this LAP is contained in Sections 7.1 through 7.4 of the XR SERIES TEACH PENDANT, Version 2.0 manual contained in Appendix A of this notebook.

LEARNING ACTIVITIES

Note: A program developed in this LAP can be used as a beginning point for the activities of LAP #13.

The activities of this LAP are meant to familiarize you with proper use of the LEARN mode of the teach pendant for the robot furnished in the High Technology Mobile Laboratory. For this LAP you should place the Table Plate #I on the grid table and secure it in position by turning the four latches, one near each corner of the grid table. Plate #I has a total of 51 one-inch squares drawn on its surface. Any of these squares may be used as a "pick-up", an "intermediate set-down", or a "final" position for a one square inch based block (labeled A) available in the Laboratory. This block is to be the payload, or object to be grasped and moved, in this LAP.

1. Prepare the robot system for use by following steps 1 through 3 of LAP #4.
2. Place the block, labeled A, on a chosen pick-up square on Table Plate #I.
3. Now perform steps B-1 through B-5 of LAP #11 to establish a soft home about two inches directly above block A at its pick-up location.
4. Next, use the teach pendant's LEARN mode, as outlined in steps B-6 through B-10 of LAP #11, to move the block through a variety of motions, to one or more "intermediate set-down" positions, and/or to a "final" position. As a "final" position you may wish to have the robot drop the block in your hand.

   Several precautions are worth mentioning for this step.

   a) When making the first movement from your established soft home, it is advisable to press the 1 key on the teach pendant so the gripper comes down slowly to grasp the block.
   b) It is advisable that you ENTER each move separately, rather than several moves simultaneously which will be the case in a later LAP.
   c) Keep the movements simple and few in number for your first attempt. After some practice you can always create more complex movement patterns. To change the program for a more complex pattern, press the RESET button and start again

with step 3 of this LAP.

d) Remember that you can cause the robot to go through single cycles by pressing the END key just after it has started, and that pressing the RUN key will start it again.

5. When finished, carry out the shut-down procedures outlined in steps C-1 and C-2 of LAP #11.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT:	ROBOTICS
LEARNING ACTIVITY PACKET (LAP) NUMBER:	13
LAP TITLE:	Teach Pendant: "EDITing" a LEARNed Program
PREREQUISITE LAP (S) :	4, 11, and 12

LAP PURPOSE: To provide information on and experience in using the teach pendant's EDIT function to modify a previously LEARNed sequential movements pattern for the robot.

COMPETENCIES TO BE DEVELOPED:

1. Describe the proper sequence of teach pendant commands needed to modify an existing sequential moves program.
2. Choose and define specific modifications to any existing sequential moves program and input the proper teach pendant commands to cause the robot to execute that modified program.

BACKGROUND INFORMATION

Note: The learning activities of this LAP should be done in conjunction with those of LAP #12.

Introduction

The EDIT mode of a teach pendant is another important feature; probably the next most important to the LEARN mode. Often when a sequence of robot movements has been LEARNed, either minor changes or additional moves must be added to the original program. Using the EDIT mode is much easier than rewriting, or reteaching, the total program. Also, for many industrial robot applications an original program may be written off-line (meaning off-line programming, on a seperate but compatible computer) and then loaded into the teach pendant. Then the teach pendant's EDIT mode is used to "trim" the robot's movements to exactly those needed. This saves valuable time because the down-time of the robot is reduced.

Editing

Use of the EDIT mode of a teach pendant on the robot furnished in the High Technology Mobile Laboratory requires several specific conditions and sequences. Some of these are as follows:

1. One or more moves must be LEARNed and ENTERed before any editing can take place. If you have not entered any moves the display will show the words nO dAtA. Also, the END key must have been pressed to signify that the entire program has been entered.
2. Pressing the EDIT key, which causes an "E" on the display, is accomplished only from the PLAY mode, indicated by a "P" on the display. Recall that a "P" appears after the END key is pressed.
3. Just after pressing the EDIT key, the display will briefly show "E" (mode) and "1" (robot movement number). Then it will show "E", "H" (motor), + or - (direction), and a number indicating the steps moved.

   Note: Neither the "H" nor the "G" motors are used on the furnished robot. Therefore, these two motors' readouts are to be disregarded.
4. Each press of the FWD (forward) key causes display of the next successive motor's data, in the sequence G, F, E, D, C, B, and A. However, the A motor's display will show only the condition of the gripper fingers, i.e., open or closed.
5. Pressing the BACK key enables one to go backward through the motor sequence.

6. The FWD and BACK keys only select a motor within one ENTERed robot movement. However, more than one motor might have been moved for the same ENTER when the program was written. These keys do not cause change from one robot movement to another.
7. Pressing the STEP key causes change from one robot movement to the next. Thus, pressing STEP should display "2" (robot movement number), following the robot movement sequence begun in step 3 above.
8. To make a change of move distance and/or direction of a particular motor:
    a) Be certain you have STEPped to the robot's movement where the change is needed.
    b) Be certain you have used the FWD or BACK keys to display the data of the motor of concern.
    c) Be certain of the direction of change that is needed.
    d) Make the change by pressing the appropriate move key, for which pressing the 10 or 1 key to cause a slow movement may be helpful.
    e) Do not ENTER the change! It will be entered automatically upon pressing the STEP key, which cuases change to the next robot movement.

Additional Information

More information on various features of the EDIT mode is contained in Sections 8.1 through 8.5 and pendant displays are explained in Section 13 of the XR SERIES TEACH PENDANT, Version 2.0 manual contained in Appendix A of this notebook.

LEARNING ACTIVITIES

Note: The last program developed and used in LAP #12 may be used as the beginning point for this LAP's activities.

The activities of this LAP are meant to familiarize you with the proper use of the EDIT mode of the teach pendant for the robot furnished in the High Technology Mobile Laboratory. Again, as in LAP #12, you will need the Table Plate #I and a block, labeled A, as the object to be moved.

The following steps assume that the last program developed and used in LAP #12 is still in memory of the the teach pendant. If not, go back to the procedures in LAP #12 and create a soft home and a series of movements to move block A.

1. Decide which robot movements need, or you want to use, for practice EDITing. One or two movements will provide sufficient practice.
2. Follow the editing procedures 2 through 8, in the Background Information of this LAP, as needed, to edit one or two of the original robot movements.
3. Press the RUN, and END, keys to cause the robot to cycle through the new (edited) movements pattern.
4. If you wish to do more robot movements editing, repeat steps 2 and 3 above.
5. When finished, carry out the shut-down procedures outlined in steps C-1 and C-2 of LAP #11.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 14

LAP TITLE: The Robot Drive Train and Drive-Motor Steps

PREREQUISITE LAP (S) : 4 and 11

LAP PURPOSE: To provide information on and experience in using the teach pendant to "step" the robot's elements through movements and determine the steps needed to make those movements.

COMPETENCIES TO BE DEVELOPED:

1. Determine the relationship between the encoder disk sections and the one-step-per-press of a move key on the teach pendant.
2. Determine the steps required to produce one revolution of drive-motor output shart.

BACKGROUND INFORMATION

Introduction

Most robot systems use some form of feedback toward achieving control of the robot's actions. Feedback may be looked upon as a signal or data which is fed back to a controller from the machine (robot) which is being controlled. The fed back signal "tells" the controller whether or not the movements of the robot were of the exact amount sent out by the controller. If there is a difference (error) between the command signal and the robot's response, the controller will act toward correcting that error.

Feedback systems can be very complex. They can involve a wide variety of sensors, one or more amplifiers, signal conditioners, and a variety of corrective circuits meant to control acelleration/decelleration, overshot, etc. These concepts are not covered in this LAP, but can be reviewed in most any text on automatic controls.

Feedback for the HTML Robot

The robot furnished in the High Technology Mobile Laboratory uses a rather simple form of feedback control. Each of the six drive motors on this robot has an encoder disk attached to the motor shaft. These disks appear similiar to the incremental encoder disk shown in Figure 1. There are other types of encoder disks, however.

As the disk rotates, the open (transparent) slots permit light to pass from a small light source to a light detector on the opposite side of the disk. When the solid (opaque) part is between the light source and detector, no light passes through. Therefore, each combination of open and solid section cause one pulse of a square wave to be sent back to the controller. Such a square wave is shown in Figure 2. Also, each revolution of the disk and motor causes a number of pulses equal to either the open or solid sections on the disk.

Each disk on the robot, and that shown in Figure 1, has two concentric rings of open and solid sections. Notice that the sections of the two rings are offset from one another. Using two rings causes two signals to be fed back to the controller. Two such signals are drawn in Figure 2. By detecting which signals leads the other, the controller can determine the direction of rotation of a motor. The robot's motors are powered by direct current. Therefore, their direction of rotation is determined by the polarity of current sent to them by the controller.

Teach Pendant Distance Keys

The teach pendant of the robot furnished in the High Technology Mobile Laboratory has three "distance" keys. They are labeled "100", "10", and "1". Their purpose is to provide a means of controlling the number of steps which a motor will rotate each time a move key is pressed. Therefore, they control the distance moved each time a move key is pressed.

The "10" key, and especially the "1" key, enable the end effector to slowly approach a position. Using the "100" key may not make this visible because the effected motor will rotate too fast. However, if the "1" key is used and a motor's move key is pressed you should see that motor rotate the distance of one open plus one closed section of the encoder disk.

Drive-Motor Gear-Boxes

Each motor on the robot has an attached gear-box. The function of these gear-boxes is to reduce the speed (RPM) and increase the torque of the motors themselves. The increased torque enables the motors to increase the force delivered to the robot's segments. The decreased speed causes smaller increments of robot segment movement for each open plus solid section of the encoder disk. This, in turn, causes greater accuracy of movement of the robot's segments.

Additional Information

More information on the "distance" keys' function is contained in Section 6.2 of the XR SERIES TEACH PENDANT, Version 2.0 in Appendix A of this notebook. Information on the robot's encoders can be found in Sections 16 and 17 of "Hands-On-Introduction to ROBOTICS, The Manual for the XR-Series Robots."

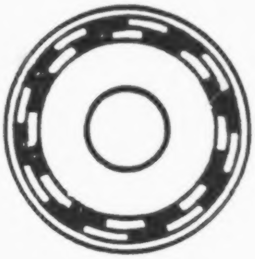


Figure 1. Incremental-Type Encoder Disk Having Two Concentric Rings of Open and Solid Sections.

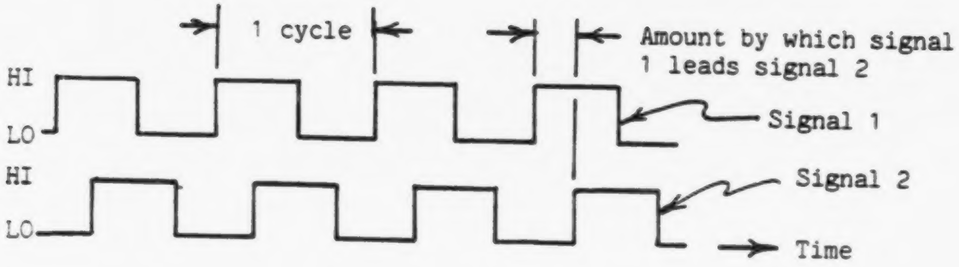

1 cycle
Amount by which signal 1 leads signal 2
HI
LO
Signal 1
HI
LO
Signal 2
Time

Figure 2. Signals Which Result From Incremental-Type Encoder Disk shown In Figure 1.

LEARNING ACTIVITIES

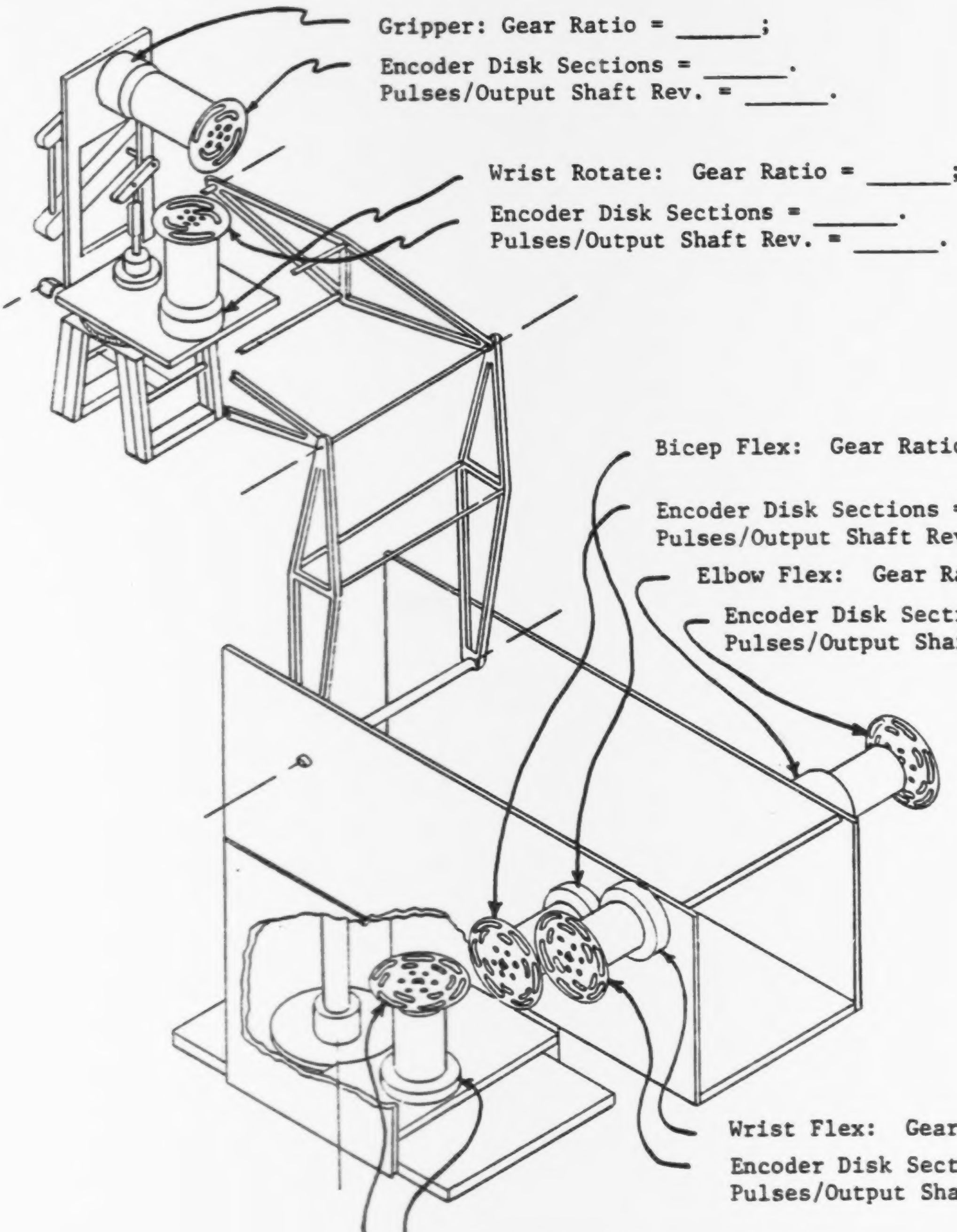
The activities of this LAP are meant to familiarize you with the encoder disks and the distance keys used on the robot furnished in the High Technology Mobile Laboratory. It is assumed that you already know how to cause the robot to make basic movements.

1. Prepare the robot system for use by following steps 4 through 7 of LAP #4.
2. Press the appropriate move keys to cause the robot's arm and hand assembly to extend to a position for easy viewing of encoder disks. Horizontal extension of the arm and hand assembly above the "+X" axis is suggested.
3. Figure 3, on a following page, is a sketch of the robot's major segments or parts. You are to do the following:
    a) View each motor's gear-box and read and record in the appropriate blank of Figure 3 the gear ratio which is stamped on the gear-box.
    b) View each motor's encoder disk and record in the appropriate blank of Figure 3 the number of sections (open or solid) in one of the concentric rings of the encoder disk.
    c) Using the formula :

       Pulses per Output Shaft Revolution = Gear Ratio x Number of Encoder Disk Sections

       Calculate the number of encoder circuit output pulses which will be produced per each revolution of the gear-box output shaft for each motor and record each in the appropriate blank in Figure 3.
4. Now, with the robot ready for use, press the "1" key on the teach pendant and then momentarily press each of the six move keys, one after another. While doing this, count the number of times you need to press each move key to cause one revolution of the corresponding encoder disk. This should equal the number of sections (open or solid) on that disk. Are these equal in each case? If not, explain why not.
5. When finished with these activities, perform the shut-down procedures of step 11 in LAP #4.

BEST COPY AVAILABLE



Gripper: Gear Ratio = ______;

Encoder Disk Sections = ______.

Pulses/Output Shaft Rev. = ______.

Wrist Rotate: Gear Ratio = ______;

Encoder Disk Sections = ______.

Pulses/Output Shaft Rev. = ______.

Bicep Flex: Gear Ratio = ______

Encoder Disk Sections = ______.

Pulses/Output Shaft Rev. = ______.

Elbow Flex: Gear Ratio = ______;

Encoder Disk Sections = ______.

Pulses/Output Shaft Rev. = ______.

Wrist Flex: Gear Ratio = ______.

Encoder Disk Sections = ______.

Pulses/Output Shaft Rev. = ______.

Waist Rotate: Gear Ratio = ______;

Encoder Disk Sections = ______.

Pulses/Output Shaft Rev. = ______.

Figure 3. Sketch Showing Encoder Disks and Drive-Motor Gear-Boxes on Laboratory Robot.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 15

LAP TITLE: Robotics Feedback, Optical Encoders And Drive-Motors

PREREQUISITE LAP (S) : 4, 12 & 13

LAP PURPOSE: To provide information on the principles of feedback and control and experience in using it toward understanding the robot's operations.

COMPETENCIES TO BE DEVELOPED:

1. Define/describe basic terms and functions associated with feedback control as found in robot applications.
2. Experiment with the optical encoders and drive-motors used on various elements of the robot.

BACKGROUND INFORMATION

Introduction

The teach pendant for the robot furnished in the High Technology Mobile Laboratory can furnish information regarding the number of steps required to cause robot segments' movement distances. LAP #14 explained that each section (open plus closed part) of an encoder disk constitutes one step of the motor. Also, one pulse is generated by each such section, and these pulses are fed back to the controller to permit accuracy of robot segment movements.

Movement Distances and Displays

All robots which use optical encoders provide a means of determining a robot segment movement distance per each step of a drive motor. However, for a revolute axis (anthropomorphic) type robot the relationship between movement distance and number of steps is linear only for the angle of movement, not for a straight-line movement.

For the Laboratory robot, straight-line movement (in the X and/or Y directions) is a result of waist, bicep flex and/or elbow flex rotations. Therefore, the distance of straight-line movement of the hand assembly depends upon the angles which the bicep makes with the base and which the bicep makes with the forearm. A non-linear relationship also exists between straight-line movement resulting from both wrist flex and elbow flex and the number of steps which cause that movement.

The number of steps required to move a robot segment a particular distance can be determined by:

1. Setting an axis at some beginning angle and reading and recording the teach pendant display of the beginning number of steps for the motor of concern.
2. Moving the robot segment which flexes about that axis to another angle; for example, 30 degrees away from the beginning angle.
3. Reading and recording the teach pendant's display of the new number of steps for the motor of concern.
4. Dividing the difference of the two steps readings by the difference of the two angles.

Note: The difference means the total number of steps made and the total degrees of angle flexed. For example, suppose that the beginning number of steps was 250 at an angle of 35 degrees and the new number of steps was -75 at an angle of -15 degrees. In this case the

$$\text{Steps/Degree} = \frac{250 - (-75)}{35 - (-15)} = \frac{325}{50} = 6.50$$

A straight-line movement which involves two or three flexed axes is not so simply solved. The steps per inch of movement distance, for example, could be determined using trigonometry. Information on such solutions are contained in some texts on robotics.

Additional Information

More information on the topics of this LAP are contained in Sections 7.1 & 7.3 of the XR SERIES TEACH PENDANT, Version 2.0 manual contained in Appendix A of this notebook.

LEARNING ACTIVITIES

The activities of this LAP will enable you to determine the number of motor steps per degree of flexing angle for several axes and the number of steps required of two motors to cause a straight-line movement distance on the robot furnished in the High Technology Mobile Laboratory. It is assumed that you already know how to cause the robot to make basic movements. Also, you should know how to use the FWD, BACK, and STEP keys to read motor data, as was covered in LAP #13.

1. Prepare the robot sytem for use by folloing steps 4 through 7 of LAP #4.
2. Press at least the waist rotate, bicep flex, and elbow flex keys to cause some movement. Then press the RESET button to clear memory.
3. Do a hard home by pressing the SHIFT and the GO HOME keys simultaneously. This is necessary prior to establishing the soft home position described in the following step.
4. Now press the appropriate keys:
   - a) To cause the gripper to close.
   - b) To cause the waist rotate, bicep flex, elbow flex, wrist flex, and wrist rotate to move to where the junction of the closed gripper fingers are just above the grid table and perpendicular to the "+X" axis at the 10" mark. This should be done very accurately, so using the "1" distance key for slow movement may be helpful!
   - c) To establish the position described in b above to be the Soft Home, i.e., press SET HOME.
5. Press the LEARN key. This will permit the robot to learn the four moves described in steps 6 through 9 below.
6. To make move #1, use the bicep flex and the elbow flex keys to cause the junction of the closed gripper fingers to move outward along the "+X" axis until it is at 16", 17", or 18" line on the grid table. Then press ENTER.
7. As move #2, use only the waist rotate move key to swing the junction of the closed gripper fingers to be centered above the -20°, -30° or -40° line on the grid table. Then press ENTER.

8. For move #3, press only the bicep flex move key to cause the bicep (upper arm) to rise to an angle of approximately 10°, 20°, or 30° above the horizontal. Then press ENTER.
9. As move #4, the last move, press only the elbow flex move key to cause the forearm to rise to an angle of approximately 10°, 20°, or 30° above the horizontal. Then press END. Doing so should cause the robot to go to the Soft HOme established in steps 4-b and 4-c above, hopefully without coming down hard on the grid table's surface!

The soft home, the one mult-axis move (#1) and the three single-axis moves which you just programmed form the positions for reading teach pendant displayed motor steps and the straight-line and angular movement distances. The data obtainable from these position readings can be used to determine relationships between motor steps and movement distances.

10. Now use the procedures outlined in the notes 3 through 7 in the Background Information of LAP #13 to read, and record in the spaces below, the proper motor step displays involved in the various robot movements. Also read, and record in the spaces below, the beginning and final straight-line distance and the angles of axis movements.
    a) At the Soft Home position:
        1) The beginning of the straight-line distance is ______ inches.
        2) The beginning bicep flex drive motor (E) number-of-steps display is ______.
        3) The beginning elbow flex drive motor (D) number-of-steps display is ______.
    b) At the end of move #1:
        1) The final position of the straignt-line movement is ______ inches.
        2) The final bicep flex drive motor (E) number-of-steps display is ______.
        3) The final elbow flex drive motor (D) number-of-steps display is ______.

Using the data from a-1 through a-3 and from b-1 through b-3 above, comment on the relationship of the total straight-line movement distance and the number of steps moved by the two motors. Then continue with the remaining end of move #1 readings requested below.

4) The begninning waist rotate angle, which should be 0° at the "+X" axis on the grid table, is ______ degrees.

5) The beginning waist rotate drive motor (F) number-of-steps display is ______.

c) At the end of move #2:

1) The final waist rotate angle, from the grid table, is ______ degrees.

2) The final waist rotate drive motor (F) number-of-steps display is ______.

Using the data from b-4 and 5 and from c-1 and 2 above, calculate the number of steps per degree rotation at the waist rotate axis. Then continue with the remaining end of move #2 reading requested below.

3) The beginning bicep flex angle (on protractor) is ______ degrees.

d) At the end of move #3:

1) The final bicep flex angle (on protractor) is ______ degrees.

2) The final bicep flex drive motor (E) number-of-steps display is ______.

Using the data from b-2, from c-3, and from d-1 and 2 above, calculate the number of steps per degree rotation at the bicep flex axis. Then continue with the remaining end of move #3. readings requested below.

3) The beginning elbow flex angle (on protractor) is ______ degrees.

4) The beginning elbow flex drive motor (D) number-of-steps display is ______.

e) At the end of move #4:

1) The final elbow flex angle (on protractor) is ______ degrees.

2) The final eblow flex drive motor (D) number-of-steps display is ______.

Using the data from d-3 and 4 and from e-1 and 2 above, calculate the number of steps per degree rotation at the elbow flex axis.

11. Briefly discuss possible reasons for any variations which may exist between the number of steps per degree rotation calculated for the waist rotate, bicep flex, and elbow flex axes.

12. When finished with step 11 above, press the RESET button on the teach pendant. This will remove all the learned moves and the soft home from memory.
13. When finished using the robot, complete the shut-down procedures of step 11 of LAP #4.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 16

LAP TITLE: Robotics End-Effector Placement Specifications

PREREQUISITE LAP (S) : 4, 11, and 12

LAP PURPOSE: To provide information on and experience in determining various end-effector placement specifications of a robot.

COMPETENCIES TO BE DEVELOPED:

1. Define/describe basic terms and the associated significance of robot end-effector placement specifications.
2. Determine the resolution and repeatability of end-effector placement of the robot.

BACKGROUND INFORMATION

Introduction

For many robot applications the end-effector must move to very specific locations within the robot's total cycle of movements. For examples, accurate end-effector placement might be necessary when the robot is used for assembly of parts or for loading and unloading of parts on a near-by machine such as a lathe. In other applications, such as grinding or deburring operations, the robot might have to move the end-effector by very small increments. These two examples indicate that end-effector placement can be very important in the performance of a robot.

End-Effector Placement Specifications

Three end-effector placement specifications can be of concern for a robot. Each of the three can be considered to inter-relate with the others. The three are described as follows:

1. Accuracy - A measurement of the difference between the end-effector taught location point, due to the command from the controller, and the played-back location. Accuracy will depend on the precision of manufacture of segments or parts of the robot itself, the feedback control system, and possibly on the accuracy of the command.
2. Repeatability - A measurement of the difference between an original played-back position and those of subsequent cycles. In other words, the extent to which the end-effector comes to the same location time after time. Thus repeatability can be effected by variations of payload and velocity and by the precision of manufacture of the robot.
3. Resolution - The smallest increment (distance) by which the end-effector can be moved. This specification is not often used for two reasons. First, the accuracy specification is more meaningfull and useful. Second, for a revolute axis (anthropomorphic) robot only the increments of angular movements are constant. Any straight-line movement involves two or more joints, which causes straight-line movements to depend upon the angles at which those joints are being used. This concept received coverage in LAP #15.

LEARNING ACTIVITIES

The activities of this LAP will provide you with information regarding the angular resolution and the repeatability of the robot furnished in the High Technology Mobile Laboratory. The robot's accuracy will not be determined because of the rather complex mathematics that would be involved in predetermining a "taught" position for the robot. It is assumed that you already know how to cause the robot to make basic moves and that you are familiar with the LEARN mode of the robot's teach pendant.

The following activities are divided into two parts; determining angular resolution, and demonstrating the concept of repeatability. Follow the instructions carefully and in the order given!

A. Resolution Determination

1. Prepare the robot system for use by following steps 4 through 7 of LAP #4.
2. Extend the gripper to a convenient position and open the gripper. Obtain the block having the rounded-end pin and place it between the fingers with the pin pointing outward from the fingers. Then close the gripper so its finger tips (the ½" thick cross bars) are in the grooves on opposite sides of the block.
3. Press the waist rotate, bicep flex, elbow flex, and wrist flex move keys on the teach pendant so the rounded-end pin is vertical, the pin's tip is just short of touching the grid table (use "1" key for slow downward movement), and the pin tip is somewhere between the 14" and 17" marks directly over the "+X" axis, which is the 0° line, of the grid table.
4. Observe and record the following beginning data:
   a) The waist rotate angle (should be 0° on the grid table) is ______ degrees.
   b) The bicep flex angle (on protractor) is ______ degrees.
   c) The elbow flex angle (on protractor) is ______ degrees.
5. Be careful that the rounded-end pin's tip does not scratch the grid table! Press the "1" distance key to permit one step per press of a move key! Then do the following:
   a) Momentarily and repeatedly press the counter clockwise waist rotate move key while counting the steps required to swing the pin through an exact angle, such as 10° or 15°, as marked on the grid table.

The number of steps in this move was ______.

The angle through which the waist flexed is ______ degrees.

b) Momentarily and repeatedly press the upward bicep flex move key while counting the steps required to raise the bicep (upper arm) through an exact angle, such as 10° or 15°, on the bicep axis protractor.

The number of steps in this move was ______.

The angle through which the bicep flexed is ______ degrees.

c) Momentarily and repeatedly press the upward elbow flex move key while counting the steps required to raise the forearm through an exact angle, such as 10° or 15°, on the elbow axis protractor.

The number of steps in this move was ______.

The angle through which the elbow flexed is ______ degrees.

6. Leave the robot powered and in this position until proceeding with the Repeatability Demonstration section later.

Using the formula:

$$\text{Resolution, degrees} = \frac{\text{Angle moved, degrees}}{\text{Number of steps in the move}}$$

a) Calculate the resolution of the waist rotate drive.

b) Calculate the resolution of the bicep flex drive.

c) Calculate the resolution of the elbow flex drive.

7. Which of the three axes' resolution was the best? The second best? The poorest?

B. Repeatability Demonstration

1. Obtain the Table Plate #II, available in the Laboratory, and place it on top of the grid table. Secure it in place by turning the four latches, one near each corner of the table.
2. Obtain a 3" x 5" card or piece of flat paper and tape is securely onto the front protruding (frame shaped) edge of the target holder which is at the back of Table Plate #II.
3. Open the robot's gripper and remove the rounded-end pin block.

4. Obtain the sharp-ended pin block and secure it in the gripper by the same procedures as used in A-2 above.
5. Press the RESET button on the teach pendant to clear the memory. Then do a hard home by pressing the SHIFT and GO HOME keys simultaneously. This is necessary prior to establishing the soft home position described in the following steps.
    a) Press the bicep flex, elbow flex, and wrist flex move keys to cause the bicep (upper arm) to be slanted at about 45° upward to your left, the forearm to be vertically downward from the elbow axis, and the hand assembly and sharpened-end pin to be horizontal.with the pin pointing toward the target.
    b) Press the waist rotate move key to cause the entire robot arm to line up directly with the "+X" axis. This is very important so that later the sharpened-end pin can pierce the target near its center!
    c) Establish this position as soft home by pressing the SET HOME key.
6. Press the LEARN key. This will permit the robot to learn the moves by which the sharpened-end pin will pierce and retract from the target.
7. Press the bicep flex and elbow flex move keys to cause the pin to approach to within about one inch from the target's center.
8. Press the "1" distance key to cause slow movement and then the bicep and elbow flex move keys some more as needed to cause the pin to move in a straight line and pierce through the target about one-half inch. Then press ENTER.
9. Now press the "1" distance key again, and then the bicep and elbow flex move keys to cause slow and careful retraction of the pin without enlarging the hole in the target. When the pin is out of the target about on-half inch, press END. This will cause entry of this last move and the robot to go to the soft home established in step B-5-c above. The robot is now programmed to try to cause the pin to go through the same hole in the target.
10. Check the repeatability for about three cycles of robot movements by pressing RUN, and then END as soon as the robot begins its third cycle.

11. Comment on how well the robot performed in these three cycles. Was the original hole enlarged? Were additional holes punched in the target? If so, what factor(s) contributed to its lack in performance?

12. When finished with the above, press the RESET button on the teach pendant. This will remove all the learned moves and the soft home from memory.
13. Open the gripper and remove the sharpened-end pin and block. Return both pin blocks and the Table Plate #II to their original locations in the Laboratory.
14. When finished using the robot, complete the shut-down procedures of step 11 in LAP #4.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 17

LAP TITLE: Robot Pick-And-Place (PNP) Operations

PREREQUISITE LAP (S): 4, 11, 12 and 13

LAP PURPOSE: To provide information on pick-and-place robotics functions and applications and experience in designing and using a pick-and-place program for object(s) movement(s).

COMPETENCIES TO BE DEVELOPED:

1. Define/describe basic terms and potential applications for robotics pick-and-place operations.
2. Design a simple pick-and-place routine for the robot.
3. Use the teach pendant's LEARN and EDIT modes in programming to cause the robot to execute a pick-and-place operation.

BACKGROUND INFORMATION

Introduction

Robots can be classified by the function(s) which they perform. One such classification is the "pick-and-place" (PNP) function. Early robots, of the 1950s and 1960s, could perform this function. But, as seen in the task classifications table of Figure 2 in LAP #2, the more sophisticated medium- and high-technology robots can perform these and many additional functions as well. A pick-and-place robot operates to pick up an object (payload), most often using a gripper-type end-effector, and to move that object to a new location. The path which the end-effector follows between pick-up and set-down of the object will depend upon the particular application being considered.

Applications

The applications for which a pick-and-place robot is suited, or the tasks which it can perform, include:

1. Machine loading and unloading. Examples could include the insertion of parts into and retrival of parts from a lathe, a milling machine, a punch press, or a forming press. Simple assembly of parts is also possible, where a part is taken from a loading chute or tube and placed in a particular location in or on another part.
2. Palletizing or stacking of parts. An example would be the placement of parts into a shipping carton as that carton moves along a conveyor belt.
3. General materials handling tasks, such as the movement of a die, a jig, or a fixture which holds a part on which operations are performed.

The High Technology Mobile Laboratory is equipped with a revolute axis (anthropomorphic) robot. Therefore, keep in mind that this robot can perform tasks of more complexity than those typically associated with a pick-and-place function.

LEARNING ACTIVITIES

Note: The program which you develop in this LAP's activities may be saved on disk for future use or demonstration. To save a program on disk, read the instructions given and cited in Part A of LAP #20.

The activities of this LAP are meant to familiarize you with the development or design and the teach pendant programming for a pick-and-place movements routine for the robot furnished in the High Technology Mobile Laboratory. Your main objective in these activities should be to plan and design the task to be performed. The secondary objective should be to program the teach pendant to cause the robot to perform the planned task's movements routine.

It is assumed that you already know how to cause the robot to make basic movements. It is also assumed that you are familiar with using the LEARN mode of the robot's teach pendant.

A. Preporatory Steps:

1. Obtain the Table Plate #I (available in the laboratory) and position it atop the grid table. Secure it in place by turning the four latches, one near each corner of the grid table.
2. Obtain one one-square-inch based block (labeled A) or one one-inch by three-inch based block (labeled B) for use as the object to be moved in these activities.
3. The Table Plate #1 has a total of 51 one-inch squares, any one of which may be chosen as the pick up or the set down location of the "A" block. This plate also has a total of 17 one inch by three inch rectangles which are appropriate for the "B" block. Choose the block you wish to use as the object to be moved.
4. In the space provided below identify the block you will use, the pick up position (square or rectangle) by its number, and the set down position (square or rectangle) by its number. Then plan and describe in writing, or by sketching, the path you choose for movement of the object. This writing or sketch should include all maneuvers of the block, such as height of lift, rotational movements, and X and Y distances. When finished with this plan, go on to the next section.

B. Programming The Planned Movements Routine

1. Prepare the robot for use by following steps 4 through 7 of LAP #4.
2. Press the appropriate move keys on the teach pendant to cause the end-effector to move out from the robot's base.
3. Press the RESET button on the teach pendant to clear memory. Then do a hard home by pressing the SHIFT and GO HOME keys simultaneously. This is necessary prior to establishing the soft home position described in the following steps.
    a) Having the "A" or the "B" block at the pick up position chosen in your planned movements routine, press the appropriate move keys on the teach pendant to cause the gripper to be open, the wrist rotated so the gripper fingers are parallel with sides of the block, and the gripper to be about two inches above the block.
    b) Establish this position as soft home by pressing the SET HOME key.
4. Press the LEARN key. This will permit the robot to learn the moves which you have identified in your plan.
5. Following your planned movements routine, proceed to press the proper move keys to cause the moves and ENTER the moves one at a time or in combinations which will cause the needed pick up, movements, and set down of the object. Keep in mind however, that:
    a) The gripper closing at pick up should be a single move.
    b) The object should be raised slightly above the table as a single move prior to other moves, so as not to slide the object on the table.
    c) The object should be lowered onto the table as a single move.
    d) The gripper opening at set down should be a single move.
    e) The gripper should be raised slightly after set down as a single move, which will prevent bumping the object upon going to soft home.

6. After the last move (probably 5-e above), press the END key on the teach pendant. This will cause entry of this last move and the robot to go to the soft home position established in step 3 above. The robot is now programmed to carry out the movements routine you have entered.

C. Cycling The Robot and Shut-Down:

1. To check the movements of the robot, press the RUN key. See if you can observe or notice each movement of your planned routine. You may wish to stop the robot (at soft home) by pressing END. It can be started again by pressing RUN.
2. When finished observing the robot's action, press END. Then, in the space below, write comments comparing the robot's actions with the original planned movements routine.

3. When finished with the above, press the RESET button on the teach pendant. This will remove all the learned moves and the soft home from memory.
4. When finished using the robot, complete the shut-down procedures of step 11 in LAP #4.
5. Replace the Table Plate #I and any blocks A and/or B which you used to their original locations in the laboratory.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 18

LAP TITLE: Robot Point-To-Point (PTP) Operations

PREREQUISITE LAP (S) : 4, 11, 12 and 13

LAP PURPOSE: To provide information on point-to-point robotics functions and applications and experience in designing and using a point-to-point program for robot execution.

COMPETENCIES TO BE DEVELOPED:

1. Define/describe basic terms and potential applications for robotics point-to-point movement operations.
2. Design a simple point-to-point routine for the robot.
3. Use the teach pendant's LEARN and EDIT modes in programming to cause the robot to carry out a point-to-point movements operation.

BACKGROUND INFORMATION

The previous LAP, #17, concerned the pick-and-place functional classification of robots. You may recall that the PNP robot function had only two locations of major concern, the pick up and the set down locations. How it moved from one to the other could vary, depending on the application.

In this LAP we are concerned with what is called the "point-to-point" (PTP) classification of function, which can be far more complex and have many more applications. A point-to-point robot moves from one point to another. This sounds simple, but bear in mind that the robot's total cycle may involve hundreds of points which must be reached in a designated sequence. Obviously then, the memory in which these points and movements are to be stored must have considerable capacity. Because of this, the point-to-point robot came into use when computers having sufficient memory became cost effective for robot control.

Applications

The applications for which the point-to-point robot is suited, or the tasks which it can perform, require more computer memory than the PNP robot and include the following:

1. Spot welding operations. The robot can move the spot welder tips along the adjoining parts to be welded, such as on an automobile body, and at each point stops long enough to weld one spot. The robot may even twist the tips slightly before opening so that the tips themselves make a cleaner break away from the metal.
2. Deburring operations. The robot can move the deburring tool, such as a small grinder, along edges of cast or other parts to remove sharp edges and protrusions. The tool may have to move along different edges at different angles as well as along both internal and external edges of the part.
3. Mass assembly operations. The robot, for example, may be used to insert specific integrated circuits, which are fed from different chutes, into the prepunched holes of a printed circuit board. Each integrated circuit may have to be rotated before insertion at a specific location on the board.

Many other applications such as drilling and gluing operations could be described. However, they would all involve a specific sequence of point-to-point moves and stops. The robot furnished in the High Technology Mobile Laboratory is capable of point-to-point operations. The memory of this robot's teach pendant is limited to 149 moves. This, however, is adequate for most any movements routine you may wish to perform.

Additional Information

More information which may be applicable to this LAP is contained in Sections 7.3 through 7.6 of the XR SERIES TEACH PENDANT, Version 2.0 manual in Appendix A of this notebook. You should read these sections so you are familiar with the difference between and the teaching and running of both single motor moves and multi axis moves. The "Principles" numbered 1 through 6 on pages 15 through 18 are particularily important!

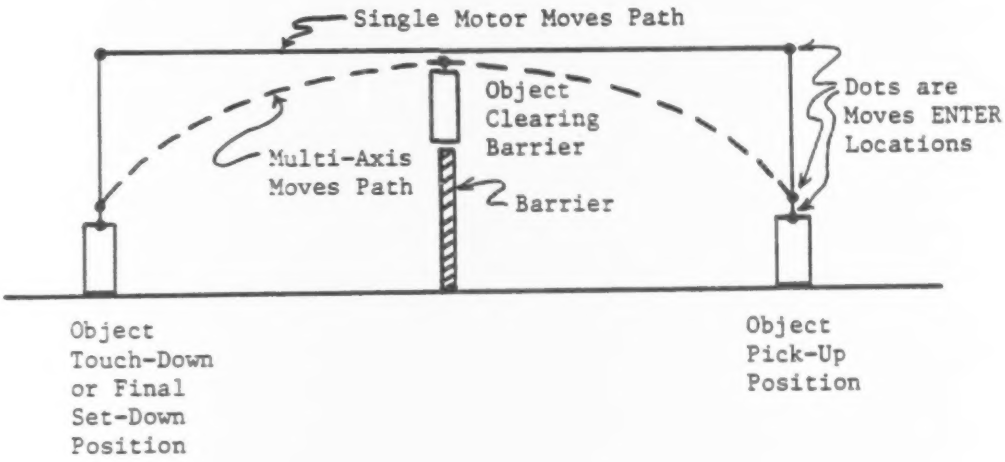

Single Motor Moves Path
Object Clearing Barrier
Dots are Moves ENTER Locations
Multi-Axis Moves Path
Barrier
Object Touch-Down or Final Set-Down Position
Object Pick-Up Position

Figure 1. Illustration of End-Effector Paths Due to Single Motor Moves and Multi-Axis Moves Programming of Robot.

LEARNING ACTIVITIES

Note: The program or programs which you develop in this LAP's activities may be saved on disk for future use or demonstration. To save a program on disk, read the instructions given and cited in Part A of LAP #20.

The activities of this LAP are meant to familiarize you with the planning, programming, and running of a point-to-point movements routine using the robot furnished in the High Technology Mobile Laboratory. Additionally, you may wish to determine the time required for programming and for running the movements routine using single moves entries and compare these with the programming and running times required for multi axis moves entries. For the same point-to-point operation, the programming time using multi axis moves entries may be longer, but the robot's cycle time should be shorter, thus indicating a time saving for robot operations. The sketch of Figure 1 on a later page illustrates the different paths which the object (payload) being moved might follow during one or more portions of a single motor moves and for a multi axis moves program.

It is assumed that you already know how to cause the robot to make basic movements. It is also assumed that you are familiar with using the LEARN mode of the robot's teach pendant.

A. Preporatory Steps:

1. Obtain the Table Plate #I (available in the laboratory) and position it atop the grid table. Secure it in place by turning the four latches, one near each corner of the grid table.
2. For this LAP's activities you are to:
    a) Use a block labeled "B" as the object to be moved.
    b) Use the even-numbered and/or odd-numbered one-inch by three-inch rectangles in Track A, Track B, or Track C as the pick up and set down locations for the object to be moved.
    c) Use two or more of the "Barriers" as obstacles across the track(s) and around or over which the movements must be made.
    d) Notice that the even-numbered and the odd-numbered rectangles within Tracks A, B, and C are sequentially turned 90° from one another. Thus, for example, if you wish to use the 11, 13, and 15 rectangles of Track B as the object's pick-up and set-down positions, the robot must rotate the object to match the rectangles' directions.
    e) Have a digital watch, or watch with second hand, for determining

the time needed to program the robot's movements and the time required for the robot to go through one cycle of movements.

3. In the space provided below draw a sketch showing the object's pick-up position, the two or more set-down positions, and where the two or more "barriers" will be located on the chosen track. Also, you should plan and list the moves which will be required of the robot. An example might begin like this: Soft Home above rectangle # with gripper open --- ; Lower Gripper to object --- ; Close gripper --- ; Raise object to clear barrier --- ; Move object to above rectangle # --- ; Rotate object CW, or CCW --- ; and so on through the final move.

List of Moves

Sketch of Object's Positions And Movements.

4. Place the object to be moved ("B" block) on the pick-up position rectangle and the barriers at the locations chosen in the plan designed and sketched in step 3 above.

B. Programming The Planned Movements Routine:

1. Prepare the robot for use by following steps 4 through 7 of LAP #4.
2. Press the appropriate move keys on the teach pendant to cause the end-effector to move out from the robot's base.
3. Press the RESET button on the teach pendant to clear memory. Then do a hard home by pressing the SHIFT and GO HOME keys simultaneously. This is necessary prior to establishing the soft home position described in the following steps:
   a) Press the appropriate move keys on the teach pendant to cause the gripper to open, its fingers to be parallel with the sides of the block, and the gripper to be about two inches above the block. Use the "1" distance key to permit slow and more accurate movements.
   b) Establish this position as soft home by pressing the SET HOME key.

Note: This is a good starting point for determining the time required for single motor moves programming of the robot. This programming starting time is __________.

4. Press the LEARN key. This will permit the robot to learn the single motor moves which you have identified in your plan of step A-3 above.
5. Following your planned movements routine of step A-3 above proceed to press the proper move keys to cause the moves and to ENTER the moves one at a time (use single motor moves only) which will cause the needed pick-up, movements, and set-downs of the object.
6. After the last move of the planned routine has been made, press the END key on the teach pendant. This will cause entry of this last move and the robot to go to the soft home position established in step B-3 above. The robot is now programmed to carry out the movements routine you have entered.

Note: The programming ending time is __________. The total time required to accomplish the single motor moves programming was __________.

C. Running The Robot With Single Motor Moves Program:

1. To check the robot's performance, press RUN and then press END as soon as the robot begins to move. If no errors such as collision with a barrier are observed, the robot's programming is all right. If

errors are present, you may either use the EDIT mode to correct them or reprogram the robot beginning with step B-3 above.

2. When the robot's performance is judged satisfactory, at a precise watch time press the RUN key and allow the robot to go through three or more cycles and begin the next cycle just before pressing END. The robot's cycle time for this single motor moves program is:

$$\text{Time per cycle} = \frac{\text{Total time for n cycles}}{\text{n cycles}} = \underline{\hspace{3cm}}.$$

3. If you do not wish to compare the above programming time and run time with that of multi axis moves programming, go to section E below.

D. Programming And Running The Robot With Multi Axis Moves Program:

Note: To complete this section is will be necessary to reprogram the robot. The objective here is to use as many multi axis moves as possible to cause the robot to do the same movements routine used in sections B and C above.

1. Repeat both step B-3-a and step B-3-b above. This will clear the memory of the old program and reestablish the soft home. This new soft home should be the same location as used for the single motor moves program.
2. Decide which moves of the original plan (see A-3 above) can be combined or changed to multi axis moves. The following hints may be helpful:
   a) The gripper closing at pickup of the object should be a single move.
   b) The object should be raised slightly above the table after pick up as a single move to avoid sliding the object on the table.
   c) The gripper opening and slight raising the gripper after the final set down should each be a single move.
   d) The most possible multi axis moves should be the rotation, the raising or lowering, and the X- and/or Y-axis movements of the object, while keeping in mind that the object must clear any barrier which you have placed on the table.

Note: The multi axis moves programming starting time is ________.

3. Press the LEARN key. This will permit the robot to learn the multi axis moves which you identified in step D-2 above.
4. Following the multi axis moves planned in step D-2 above, proceed to press the proper move keys to cause the combined or single moves as needed and to ENTER them for the robot's future actions.

5. After the last move(s) of the planned multi axis moves program has been made, press the END key. This will cause entry of the last move(s) and the robot to go to the soft home. The robot is now programmed to carry out the multi axis moves program.

Note: The programming ending time is __________. The total time required to accomplish the multi axis moves programming was __________.

6. Check the robot's performance by reading, and, if necessary, making error corrections as described in step C-1 above.
7. Determine the Time per robot cycle by repeating step C-2 above.

$$\text{Time per cycle} = \frac{\text{Total time for n cycles}}{\text{n cycles}} = \underline{\qquad\qquad}.$$

E. Questions and Robot Shut-Down:

1. Did the robot perform as you intended for the single motor moves programming? If not, what errors and/or difficulties were encountered?

2. If you completed section C, did the robot perform as you intended for the multi axis moves programming? If not, what errors and/or difficulties were encountered?

3. If you completed section C, which, the single motor moves or multi axis moves, required the most programming time? Which did you consider the most difficult and why?

4. If you completed section C, which, the single motor moves or the multi axis moves, required the most time per robot cycle? Explain what this could mean for robot efficiency?

5. When finished with the above, press the RESET button on the teach pendant. This will remove all the learned moves and the soft home from memory.
6. When finished using the robot, complete the shut-down procedures of step 11 in LAP #4.
7. Replace the Table Plate #I, the blocks A and/or B, and the barriers which you used to their original locations in the laboratory.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 19

LAP TITLE: Robot Continuous-Path (CP) Operations Demonstration

PREREQUISITE LAP (S) : 4, 11, 12, and 13

LAP PURPOSE: To provide information on continuous-path robotics functions and applications and experience in designing and using a simple simulated continuous-path program for robot execution.

COMPETENCIES TO BE DEVELOPED:

1. Define/describe basic terms and potential applications for robotics continuous-path operations.
2. Design a simple simulated continuous-path routine for the robot.
3. Use the teach pendant's LEARN and EDIT modes in programming to cause the robot to carry out a simulated continuous-path operation.

BACKGROUND INFORMATION

Introduction

The pick-and-place and the point-to-point functional classifications of robots were covered in LAP #17 and LAP #18, respectively. In both of those functional classifications the robot's movements were from or to specific points or locations within its cycle. Reaching these points in a particular sequence was the primary objective, so long as the robot cleared any obstructions, i.e., didn't bump into objects, which might be between the points.

In this LAP we are concerned with a third classification, the "continuous path" (CP) function. The unique characteristic of the CP robot is that the path which the end-effector must follow is critical, even though the end-effector still moves from one point to the next, the next, etc. The three-dimensional sketches of Figure 1 will aid in explanation of this. In both sketches let's assume that the object to be moved must reach points a, b, and c. However, the paths between these points are very different. The sketch of Figure 1-a could be that of a point-to-point function robot. Notice that a total of only nine points are used to define the path.

A far more critical path is illustrated in the sketch of Figure 1-b. Points a, b, and c are still reached and the end-effector still moves in the X, Y, and Z axes directions. In this case, however, a total of 31 points are used to define the path, and many more could be added to cause the end-effector to trace an even smoother curve. From this illustration one should realize that the more critical the path, the greater the number of points needed to accurately define it. Also, the greater the number of points needed, the larger the memory capacity needed.

Applications

The continuous path functional robot often has greater computer memory capacity and may be built for greater precision than the PNP or PTP robots. These two factors, and other possible features, enable the CP robot to handle critical-path applications. Some of these applications are:

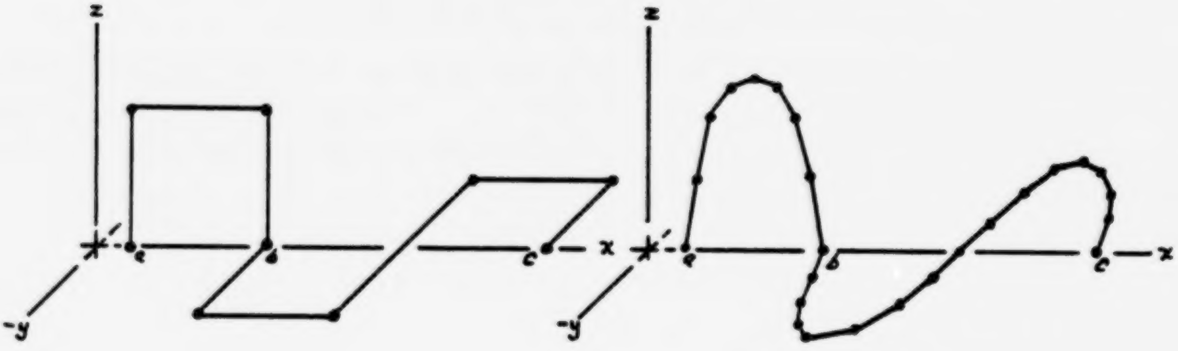
1. Paint spraying operations. For best paint application, the robot should move the sprayer's nozzle at a constant velocity, at a fixed distance from, and always at the same angle to the surface to be sprayed. Since one of the robot's major movements in this case is the base swivel or waist rotate, the robot's bicep, elbow and wrist flex, and a possible wrist yaw, need to be moved as the sprayer's nozzle sweeps past the surface.

2. Seam or arc welding. Many welds are of considerable length and they are often along a curved path. For arc welding the length of arc (distance from feed wire to parent material) is important toward determining the quality of the finished weld. Also important is the angle at which the weld bead is deposited. The continuous path functional robot can be programmed, typically using a teach pendant, to maintain the arc length and angle at the proper values.
3. Cutting operations. The acetylene torch, the laser, and other devices can be used to make both straight and curved cuts in a wide variety of materials. The pieces or parts which are cut from the stock material may be very intricate. Such cutting requires the entry into memory of many points along the path. Thus, the continuous path functional robot may be used to move the cutting tool.
4. Inspection operations. The end-effector of a robot may be equipped with a variety of sensors and then used to inspect parts. The sensors might slide along the part's surface to determine its dimensions, its smoothness, the accuracy of features on the part, or to identify the part. Signals from these sensors are sent to a computer where they are compared with one or more prescribed values. This comparison enables the computer to make decisions regarding the acceptance or rejection of the part being inspected.

Additional Information

The robot furnished in the High Technology Mobile Laboratory is capable of demonstrating the continuous path functional robot. The main limitation is that its teach pendant can store only 149 moves. However, this quantity should be sufficient for any one of a variety of continuous path demonstration routines.

The information concerning multi axis moves in Sections 7.3 through 7.6 of the XR SERIES TEACH PENDANT, Version 2.0 manual in Appendix A of this notebook will be helpful toward programming a continuous path functional routine. These sections should be read before proceeding to the learning activities of this LAP.



(a)

(b)

Dots indicate move(s) entries.

Figure 1. Illustration of a) Point-To-Point, and b) Continuous-Path Programming Entries For End-Effector Movement.

LEARNING ACTIVITIES

Note: The program or programs which you develop in this LAP's activities may be saved on disk for future use or demonstration. To save a program on disk, read the instructions given and cited in Part A of LAP #20.

The activities of this LAP will familiarize you with the programming and running of a continuous path functional robot. The teach pendant of the robot furnished in the High Technology Mobile Laboratory has memory capacity for only 149 moves. Therefore, in these activities you should use multi-axis moves entries whenever possible.

It is assumed that you already know how to cause the robot to make basic moves. It is further assumed that you are familiar with use of the LEARN mode of the robot's teach pendant and with the advantages and use of multi-axis moves entries.

A. Preporatory Steps:

1. Obtain the Table Plate #II (available in the laboratory) and position it atop the grid table. Secure it in place by turning the four latches, one near each corner of the grid table.
2. This #II Table Plate has two tracks along which the robot can move a golf ball. You may choose any of the following three routines, each requiring three-dimensional movements of the "golf club" which the robot's gripper will hold.
   a) Move (push) the ball from Pocket 1 up the inclined track until it rolls into Hole 1, and then you manually place the ball back in Pocket 1 for the robot's next cycle.
   b) Move (push) the ball from Pocket 2 up the inclined track until it rolls into Hole 2, and then you manually place the ball back in Pocket 2 for the robot's next cycle.
   c) Move (push) the ball from one pocket up the inclined track until it rolls into that track's hole, and then continue with the routine to push the ball from the other pocket up the other inclined track until the ball rolls into that track's hole. This routine (c) should be tried only after successfully running either routine a or b above, and assuming that somewhat fewer than 75 moves were needed to accomplish either routine a or b above.

3. Obtain the golf ball (available in the laboratory) and place it in Pocket 1 (see choice A-2-a above) or in Pocket 2 (see choice A-2-b above).
4. Prepare the robot for use by following steps 4 through 7 of LAP #4.
5. Press the appropriate move keys on the teach pendant to cause the end-effector to move out from the robot's base.
6. Press the RESET button on the teach pendant to clear memory. Then do a hard home by pressing the SHIFT and GO HOME keys *simultaneously*. This is necessary prior to establishing the soft home position described in steps B-1 and -2 *below*.
7. Move the end-effector out from the robot's base. Then obtain the "golf club" (available in the laboratory) and hold it in position between the cross-bars of the gripper's fingers while closing the gripper. The cross-bars should be *inside the grooves* of the golf club when the gripper is closed.

B. Programming And Running The Continuous-Path Routine:

1. Having completed step A-7 above, press the appropriate move keys on the teach pendant to cause the golf club to be positioned *a short distance behind the golf ball*, which was placed in one of the two pockets. This is more easily accomplished if you *press the "1" distance key to slow the robot's movements*!
2. When you are certain that the golf club is properly positioned, i.e., low enough to push the ball, but also not so low as to rub the track's sides, press the SET HOME key. This will establish this position as soft home within your routine of movements.
3. Press the LEARN key. This will permit the robot to learn the moves necessary to push the golf ball up the inclined track.
4. Press the "1" distance key on the teach pendant to cause slow robot movements. Then press the appropriate move keys *to cause short* (from one-inch to two-inch) *increments of pushing the golf ball* up the inclined track until it rolls into the hole at the other end of that track. In doing so, keep in mind the following:
   a) Wherever possible, *multi-axis moves* should be ENTERed, so as to save memory. *Keep a running account of the number of times you ENTER move(s)*!

   b) The surface of the golf club should be maintained perpendicular to the direction of golf ball movement.
   c) The golf club should be low enough to roll the ball and yet clear the track's side rails.
   d) The ball should be rolled only far enough to drop into the hole.

5. As soon as the ball rolls into the hole (B-4-d above) ENTER that move. Then the last move should be to raise the golf club about two inches, so it will not bump anything on its return to soft home. Then press END. This will cause entry of this last move and cause the robot to go to the soft home position described in steps B-1 and -2 above. Remember to add this last move to your account of the number of move(s) ENTERed. The robot is now programmed to carry out the continuous-path movements routine you have entered.
6. To check out the robot's performance:
   a) Place the golf ball in the Pocket of the track you are using.
   b) Press RUN and then press END as soon as the robot begins to move. This will stop the robot at soft home after one cycle. Also be ready to press the RESET button to stop the robot if any danger becomes evident!
7. If no difficulties were encountered in your check out of the robot's performance, you may cycle the robot by pressing RUN. Each time, however, you will need to place the ball in the Pocket of the track you are using. The robot may be stopped, at the soft home position, whenever you wish by pressing the END key.
8. After successfully running the continuous-path routine, answer the following:
   a) How many moves were used in your program?

   b) Which axes (X, Y, and/or Z) were involved in your movements routine?

   c) Which axis or axes was/were involved in causing the ball to move horizontally? Which was/were involved in causing vertical (up the incline) movement?

d) Briefly explain why this exercise was a demonstration of a continuous-path movements routine.

9. If you do not choose to program the robot to push the golf ball along both tracks, go on to section D.

C. Optional: Notes On Movements Routine Involving Both Tracks:

If the program you used in section B above involved fewer than half of the 149 moves of teach pendant memory capacity, you may wish to program the robot to push the golf ball along both tracks. This was given as choice c in step A-2 above.

In programming the robot to follow first one track and then the other, keep the following notes in mind.

1. Before reprogramming the robot, repeat steps A-6 and -7 above.
2. The soft home may be established a short distance behind the ball in either of the two pockets. Repeat steps B-1, -2, and -3 above to establish the soft home.
3. When the golf ball's movements along the first track have been ENTERed as robot moves (steps B-4 and -5 above) the next moves will involve moving the golf club to a short distance behind the ball in the pocket into which it has just rolled. These should be done as a series of single motor moves so as to avoid any collisions in going from one track to the other!
4. Movements along the second track are made using the same guidelines as for the first track. See step B-4 above.
5. When the ball rolls into the hole at the end of the second track, single motor moves should be used again to raise (vertical movement) the golf club and then to swing (horizontal movement) the golf club to where it is several inches above the soft home position of the first track. Then press END, which will cause entry of the last single motor move and the robot to decend to the soft home position.
6. Check out of the robot's performance follows the same procedures as given in step B-6 above. However, in this case you do not need to manually return the ball to the starting pocket.

7. You may cycle the robot by repeating step B-7 above.
8. After successfully running this two-track continuous-path routine, answer the following:
    a) How many moves were used in your total two-track program? Was this more than twice the number used in the single track program of section B? If so, why?

    b) Briefly describe the quality of performance of the robot. Did the golf club roll the ball accurately along each track? Did the golf ball have sufficient time to reach each pocket before the golf club arrived to push it along the tracks?

D. Robot Shut-Down:

1. When finished with the above, press the RESET button on the teach pendant. This will remove all the learned moves and the soft home from memory.
2. Press the appropriate move keys to raise the hand assembly well above the golf ball tracks. Then open the gripper and remove the golf club.
3. When finished using the robot, complete the shut-down procedures of step 11 in LAP #4.
4. Replace the Table Plate #II, the golf ball, and the golf club to their original locations in the laboratory.

UNIVERSITY OF
NORTH DAKOTA
DEPARTMENT OF INDUSTRIAL TECHNOLOGY

HIGH TECHNOLOGY MOBILE LABORATORY

Learning Activity Packet

MAJOR INSTRUCTIONAL UNIT: ROBOTICS

LEARNING ACTIVITY PACKET (LAP) NUMBER: 20-A, -B, and -C

LAP TITLE:
A) Robot Program Up-Loading/Down-Loading To/From A Computer
B) Robotics Languages
C) The Future of Robotics

PREREQUISITE LAP (S) : For 20-A LAPs 17, 18 or 19; For 20-B and 20-C LAP #2

LAP PURPOSE: To provide information regarding uploading and downloading of robot programs; regarding various features which are desirable in robotics languages; and regarding the future of the robotics field.

COMPETENCIES TO BE DEVELOPED:

1. Be able to upload and download a robot program to and from a computer.
2. Be able to identify desirable features of a robotics lagnuage.
3. Be able to identify possibilities in the future of the robotics field.

BACKGROUND INFORMATION

This LAP will provide information on three topics which are important to the field of robotics. Most of this information is provided in the form of reading materials. However, if you have completed LAP #17, #18 or #19 you will have noticed that those robot programs could be stored for future use. Such storage and reuse of programs provides the only hands-on activities of this LAP.

A. Robot Program Uploading and Downloading:

To upload means to transfer a robot program, which was developed using a teach pendant, from the teach pendant memory to the memory in a computer. The advantage of doing this is that the program can then be transfered again onto a disk, or on punched tape or cassettee tape, for future use.

To download means to transfer a robot program from the computer memory into the teach pendant memory. The program which is downloaded may have been used earlier and stored on disk or tape. However, the program may have been developed on a separate computer, which is called off-line programming. Use of off-line programming can save valuable robot time because the robot's teach pendant is not used for the actual programming. It may be used, however, to edit minor changes into the off-line developed program.

Instructions for uploading and downloading a program are contained in Section 9 (and on pages 40 and 41 for using the Apple IIe computer) of the XR SERIES TEACH PENDANT, VERSION 2.0 contained in Appendix A of this notebook. It is very important that you read this information prior to attempting the uploading or downloading of a program. Additionally, the following comments will be helpful:

1. Proper connection of the host computer (Apple IIe)is shown in Figure 1a on page 3 of the manual.
2. You will need a disk on which to store your program. Contact your instructor as to how to obtain one.
3. The disk labeled "Upload/Download" contains the manufacturer's upload/ download program. You should not use this disk for storage of your program(s)!
4. The Upload/Download disk, when run, will give you three choices as to what card is used in the Apple IIe. This is a number CCS 7710A card. Therefore, your coice (response) must be 2.

B. ROBOTICS LANGUAGES:

The languages used to program robots have become simpler as the state-of-the-art of robotics has progressed. Evidence of this is seen in the easy use of the teach pendant for the robot furnished in the High Technology Mobile Laboratory. This robot can be programmed simply by pressing the proper move keys to cause robot movements and then entering these moves into memory for playback at a later time. This is much simpler than writing the program on a computer, such as the Apple IIe, without having prior knowledge of what each line of the program will cause the robot to do.

A summary of what a good robot language should provide would certainly include the following:

1. Simplicity and understandability for use. The language should use commands (words) which are meaningful to the programmer. The words on the laboratory robot's teach pendant are a good example of this.
2. Speed of caused movements. The language should permit the computer to command the robot at the maximum speed at which the robot must perform. One example is that the BASIC language is too slow for driving some types of electric motors. Therefore, when such limitations are involved a faster language must be used. For some robots BASIC may be used to program the computer, and this converted into machine language within the computer to provide a faster language to run the robot.
3. Sensor interfacing. The language should readily accept high speed signals from sensors located on either or both the robot and related machines.
4. Multiple control. The language and the control circuits should be capable of providing sequenced control to two or more robots, as well as to other related machines.

Further information regarding robot languages is provided in two separate parts in the following pages of this section (B). Reading either or both of these will provide you a more thorough knowledge of what a good robot language should contain.

From: Emerson, Thomas A. Technological Advancements In Industrial Robotics, Unpublished Masters Thesis; Department of Industrial Technology, University of North Dakota; May, 1984; pp. 28-29, 31-32.

HIGH TECHNOLOGY INDUSTRIAL ROBOTS

Control Developments

The progression to third generation or high technology industrial robots was marked primarily by controller application and specialized component advancements. A significant characteristic of these robots was that their control functions were now performed by a microprocessor rather than a mini-computer. The change was due to the microprocessor being more economical than the mini-computer, yet it was also able to perform all the necessary tasks.

At this stage, the programming of industrial robots advanced from on-line to off-line methods. These procedures refer to the conditions under which a program is developed for the controller of a robot. On-line programming exists where a program is developed in the controller which is directly connected to the robot. While this is being done the robot must be taken out of production. The program is typically developed by a floor technician using a teach pendant, for point-to-point operation, where only the endpoints of each location are recorded, or by personally guiding the robot arm through the desired contour motion, for continuous path operation, while the controller reads and stores the axes' positions many times a second. This form of programming has two inherent problems. One is the significant loss of production due to the often extensive programming time. The other is the possibility of a less-than-optimal program due to the technician's lack of knowledge concerning the entire process being performed. In response to this, off-line programming was devised. This process allows a program to be developed independently of the main controller. As a result, the robot can remain in production and a person with greater knowledge of the manufacturing process and robot kinematics can develop a more efficient program on an isolated CRT terminal (12). Upon completion, the program is down-loaded into the on-board controller.

The first languages developed for off-line programming dealt primarily with the naming of desired operation motions rather than specific joint commands. One of the first successful languages was called JARS. It was based on Pascal and established to provide a series of motion commands for assembly tasks. Soon following this was the introduction of VAL. This motion language was derived from BASIC. It was an improvement over JARS in that it allowed the robot to work in moving frames of reference. This was especially beneficial for production or assembly tasks performed in conjunction with a moving conveyor. The lineage of these languages may be observed in Figure 11. After this, a language known as AML was introduced. It was developed to provide a strong base motion language for users with a wide range of experience. The significance lies in its purpose of establishing a common bond between the variety of motion languages.

One of the highest of motion command languages is known as HELP. It was developed for the purpose of controlling multiple Cartesian arms in assembly tasks. The commands are based on Pascal and allow for controlling up to four arms with a combined total of twelve axes. Being developed slightly later but somewhat parallel to these were

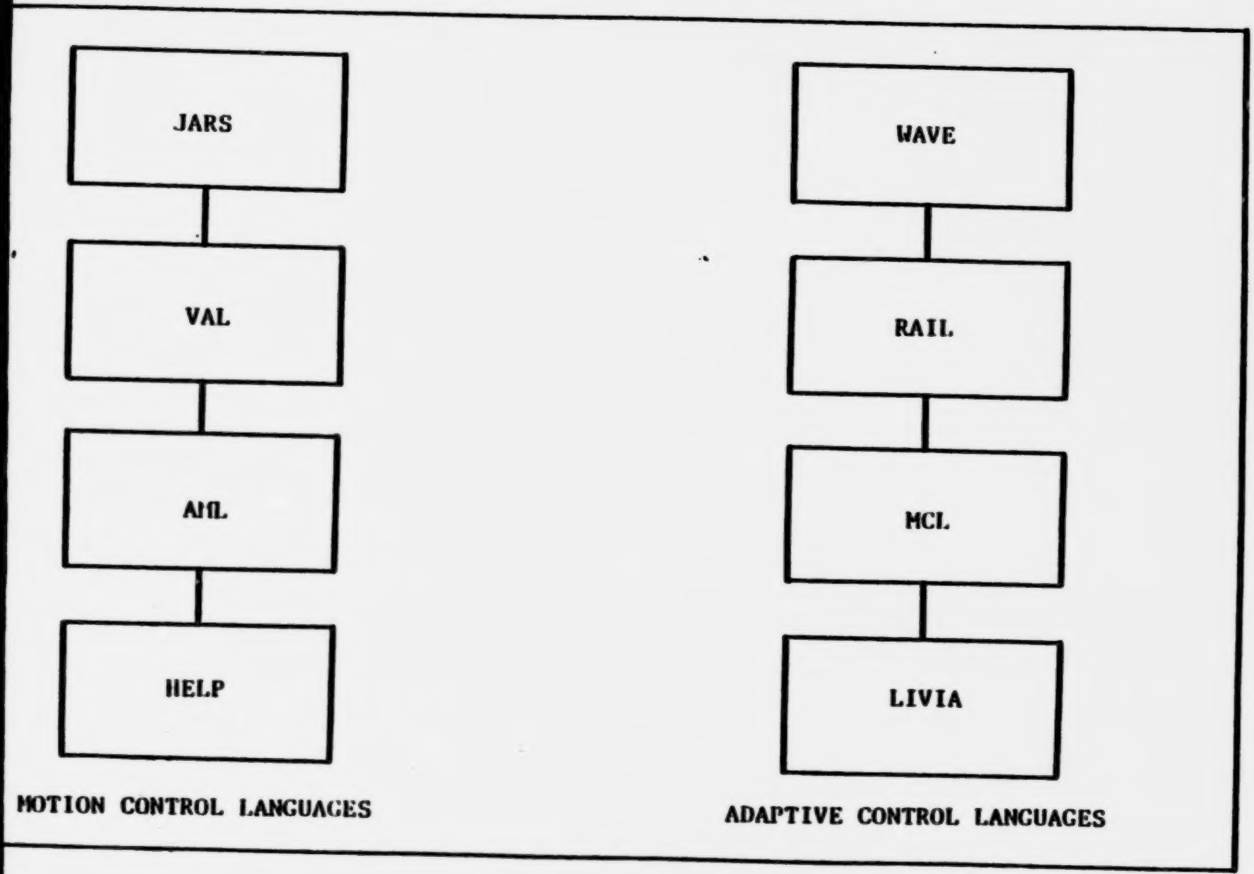

JARS
VAL
AML
HELP
MOTION CONTROL LANGUAGES
WAVE
RAIL
MCL
LIVIA
ADAPTIVE CONTROL LANGUAGES

Fig. 11. Lineage of the Common Off-Line Programming Languages.

languages incorporating motion and sensory input, which are known as adaptive control languages. The first promising one of this type was called WAVE. It successfully demonstrated that a vision system could be coordinated with the manipulator arm motions of a robot or other manufacturing devices in a work cell. A second language was called RAIL. It offered the same capabilities as WAVE, but also that of visual inspection for quality control tasks (13). The third and most advanced combination language was called MCL. Its capabilities far exceed those of previously mentioned languages. To begin with, it has the ability to control any robot through the use of a post-processing routine by the controller which changes the MCL code into a form that is compatible with the desired robot. It also has the full range ability to control all the activities of every device in a work cell, such as the robot, machining equipment, and the vision system. In addition, it has the ability to process real-time sensory data and make logical decisions based on this information. An unusual feature is its ability to specify convenient frames of reference during programming, which are interpreted by the individual cell controllers in relation to their hardware (14).

The last language of this type is called LIVIA. Its primary function is that of programming industrial vision systems. This language, along with those previously discussed, is intended for use with the binary system of vision. With LIVIA, objects are described by a set of feature values which are stored in memory. The camera data is then compared to this model for performing recognition and inspection tasks (15). A further comparison of these languages is presented in Table 1.

TABLE 1

PROGRAMMING LANGUAGE COMPARISONS

Language Type	AML	HELP	JARS	MCL	RAIL	VAL
subroutines			X			
extension				X		
new language	X	X			X	X

Ability to Control Multiple Arms	AML	HELP	JARS	MCL	RAIL	VAL
multiple arms		X		X		

Motion Types	AML	HELP	JARS	MCL	RAIL	VAL
joint interpolated	X	X	X		X	X
straight-line			X	X	X	X
continuous path						

Sensor Interfaces	AML	HELP	JARS	MCL	RAIL	VAL
vision	a	X	X	X	X	X
force	X	X				
proximity						
limit switch	X		X	X	X	X

SOURCE: Conference Proceedings, 13th International Symposium On Industrial Robots And Robots 7, Future Directions (Dearborn: Society of Manufacturing Engineers, 1983), pp. 12-63 - 12-66.

[a] Reported by the IBM T. J. Watson Research Center, Yorktown Heights, NY; not commercially available.

From: Liming, Gary "ROTREP: An Interpretive
Language for Robot Control," Byte Magazine

ROTERP: An Interpretive Language for Robot Control

High-level languages may help bridge the gap between artificial intelligence and the home experimenter's robot.

Gary Liming
1224 South Wheaton
St. Charles, MO 63301

A quick survey of the information being published on experimental robotics and artificial intelligence reveals a gap in practical information between the two subjects where there should be a bridge. Paradoxically, while most experiments in artificial intelligence are performed in high-level languages (notably LISP), most books and articles for robotics experimenters concentrate on hardware design and leave software control to assembly-language routines.

The assembly-language approach to robot control involves combining routines to produce a particular behavior in a robot. This method does have practical advantages: it requires little initial planning, it is usually memory-efficient, and the code can be entered by using simple switches or a keypad, making more expensive peripherals dedicated to the robot unnecessary. A major drawback, however, is the time and effort it takes to reorganize the routines to produce new behavior in the robot.

Perhaps the greatest disadvantage of using assembly language is the difficulty other experimenters have understanding the origins of a robot's behavior. Because assembly language is difficult to read, the underlying cause of the robot's behavior may elude even the well-versed experimenter. Duplicating the behavior of a robot with a different processor is usually even more difficult.

A microprocessor specifically designed for robotic control would resolve these problems. Such a microprocessor would use simple commands that exercise all of the robot's capabilities. A program for this microprocessor might look something like listing 1, which causes the robot to slowly walk a three-foot square. In addition to being easier to understand than assembly language, the program requires only 24 bytes of storage.

Enter the "Pseudoprocessor"

Our theoretical microprocessor would be practical only if all the components of the robot could be anticipated and mass production could justify its cost. In fact, custom processors have been used in intelligent instruments and even in sophisticated toys, but an experimental robot is a low-volume item and must be flexible enough to accommodate major new functions and devices. To expect all experimenters to use the same microprocessor is unrealistic.

Listing 1: *This example of a first attempt at a robotic language suggests how a robot-control language should look. Instructions are simple enough to be very flexible (they might be used in any number of situations) yet powerful enough so that the programmer doesn't get bogged down in details.*

```
0C30  05          SLOW
0001  08          FEET
0002  30  0003    FORWARD 3
0005  34  0090    TURN RIGHT 90
0008  30  0003    FORWARD 3
0011  34  0090    TURN RIGHT 90
0014  30  0003    FORWARD 3
0017  34  0090    TURN RIGHT 90
0020  30  0003    FORWARD 3
0023  00          HALT
```

In lieu of a custom microprocessor, we might create a "pseudoprocessor" in software. Not only would it execute our robotic instructions, but it would be easy to extend or modify as well. This pseudoprocessor could be in the form of a small interpreter, possibly in PROM (programmable read-only memory). The interpreter

Single-Board Computer
—Microprocessor
—Limited available memory
—I/O interfaces

Output Devices
—Motors for forward, reverse, and turning motions
—Numeric display
—Bell or buzzer

Input Devices
—Proximity switches or ultrasonic sensors
—Numeric keypad
—Light detector

Table 1: *Attributes of a simple experimental robot.*

would fetch each instruction of the program from memory and call an appropriate routine to execute that instruction. The pseudoprocessor technique is not new; languages like Pascal and even some adventure-type games have been implemented for microprocessor systems using the concept.

If all of this sounds like an argument on behalf of the use of high-level languages, it is. A robotic pseudoprocessor has a number of advantages. First, it provides programming at a behavioral level that is easy for the programmer (or anyone else) to understand. And it creates programs that are less machine-dependent. It is also conducive to the creation of more powerful instructions than those on most microprocessors. Finally, it can be extended and modified, if designed properly.

A Simple Design

Let's see how the pseudoprocessor approach can be applied in a simple example. Table 1 shows the attributes of a basic experimental robot. It consists of an SBC (single-board computer) like many of those currently available, a frame and motors constructed with treads or wheels for mobility, some proximity sensors and light detectors, a bell or buzzer, and a numeric display and keypad (found on the SBC).

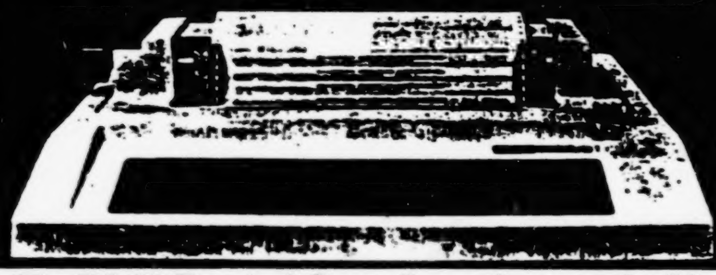
Given these simple peripheral devices, we can envision what a pseudoprocessor for a robot will look like. Of course, it will need instructions to turn the drive motors on and off and to acquire data from the sensors. It will have registers to count units of time and distance. Instructions that change the program flow and conditional testing will enhance its ability to make decisions. And it will include such standard items as a program counter, general-purpose registers, and I/O (input/output) instructions.

Figure 1 shows the architecture of the interpreter for the simple robot I named ROTERP. It has 26 general-purpose 16-bit registers, a program counter, condition codes, and some special-purpose registers. (I chose to have 26 general-purpose registers because they are easily represented by the letters of the alphabet.)

The array register, a location that points to the array being referenced, functions like an index register found on other processors. The speed register is a location for a number that the movement instructions use to determine a rate of speed. The distance-units register is a location that movement instructions use to set a scale for movement, and the time-units register specifies a time scale.

To produce a particular behavior, the registers are manipulated by the

Tune up your LA36

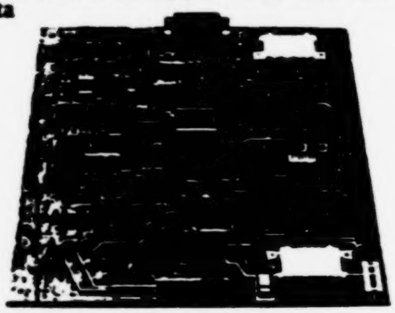


The DS120 Terminal Controller makes your LA36 perform like a DECwriter® III.

The Datasouth DS120 gives your DECwriter® II the high speed printing and versatile performance features of the DECwriter® III at only a fraction of the cost. The DS120 is a plug compatible replacement for your LA36 logic board which can be installed in minutes. Standard features include:

- 165 cps bidirectional printing
- Horizontal & Vertical Tabs
- Page Length Selection
- 110-4800 baud operation
- 1000 character print buffer
- X-on, X-off protocol
- Self Test
- RS232 interface
- 20 mA Current Loop interface
- Top of Form
- Adjustable Margins
- Double wide characters
- Parity selection
- Optional APL character set

Over 5,000 DS120 units are now being used by customers ranging from the Fortune 500 to personal computing enthusiasts. In numerous installations, entire networks of terminals have been upgraded to take advantage of today's higher speed data communications services. LSI microprocessor electronics and strict quality control ensure dependable performance for years to come. When service is required, we will respond promptly and effectively. Best of all, we can deliver immediately through our nationwide network of distributors. Just give us a call for all the details.



datasouth computer corporation
4216 Stuart Andrew Blvd. • Charlotte, North Carolina 28210 • 704/523-8500

Circle 141 on inquiry card.

BEST COPY AVAILABLE

instruction set given in table 2, which shows each instruction op (operation) code, its associated mnemonic, the type of operand it uses, and the number of bytes per instruction. Although the op codes are shown in hexadecimal, they could easily be given in decimal; hexadecimal was chosen only for convenience. The first 64 instructions control the robot's movement while the last 64 instructions control the processor registers. Because ROTERP uses 8-bit op codes, 128 possible instructions are still available for expansion.

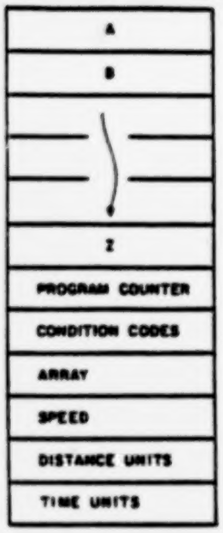


Figure 1: *Architecture of ROTERP, the interpreter for a simple robot. The interpreter handles 16-bit-wide registers (there is a general-purpose register for each letter of the alphabet as well as registers for condition codes and special purposes).*

In order to understand the origin of behavior in our simple robot, we need to look more closely at the instructions it follows. The first group of instructions, op codes 00 through 0F hexadecimal, are all 1-byte instructions. HALT simply returns control to the monitor ROM (read-only memory) on the SBC. ZERO initializes all the processor registers except the program counter. FAST, MEDIUM, and SLOW place a predetermined number in the speed register. Similarly, INCHES, FEET, and METERS store a value in the distance-units register, and MILLSEC, SECONDS, and HOURS set the time-units register. REPEAT places the starting address of the program into the program counter and normally causes it to loop indefinitely at the end of a program. RETURN signals the end of a subroutine.

The next group of instructions, 10 through 1F hexadecimal, are each 2 bytes long and use a single-byte operand to specify a general-purpose register that contains a working value. For instance, BEEP C, represented as 17 03, causes the robot to beep the number of times indicated by register C. FORWARD B (10 02) causes the robot to move forward the number of units defined in register B. Similarly, HESITATE A (19 01) causes the robot to wait A units of time, DISPLAY A displays register A, and ENTER A places a number from the keypad in register A. TEST tests the indicated register and sets the appropriate condition codes. RANDOM returns a random number and places it in the given register. SCAN and PROXIMITY read the respective

THE
BENCHMARK®

A Word Processor
Worthy of Comparison . . .

FEATURE	BENCHMARK 2.0	Wordstar 3.21	BENCHMARK 3.0
Ease of Operation	YES	NO	YES
Plain English Commands	YES	NO	YES
Use Cursor Keys	YES	NO	YES
Control Characters	NO	YES	NO
Function Keys	YES	NO	YES
Computer Aided Tutorial	YES	YES	YES
Descriptive Directory	YES	NO	YES
29 Character Title	YES	NO	YES
Author & Operator ID	YES	NO	YES
Document Size	YES	NO	YES
Creation Date	YES	NO	YES
Revision Time & Date	YES	NO	YES
Standard Editing Features	YES	YES	YES
Go To Any Page	YES	NO	YES
Interactive Printing	YES	NO	YES
Automatic Reformatting	YES	NO	YES
Automatic Repagination	YES	NO	YES
Headings and Footings	YES	YES	YES
Multi-line	YES	NO	YES
Keyboard Phrases	YES	NO	YES
Proportional Printing	NO	NO	YES
Business Graphics	NO	NO	YES
Paragraph Assembly	NO	NO	YES
Edit Marking	NO	NO	YES
Auto Widow/Orphan Protect	NO	NO	YES
Auto Footnoting	NO	NO	YES
Built In Calculator	NO	NO	YES
Price	$249	$495	$499

DEMAND A COMPARISON!

Call or Write

Metasoft Corporation 711 E. Cottonwood, Suite E, Casa Grande, AZ [illegible]

Circle 286 on inquiry card.

Op code	Mnemonic	Operand type	Bytes required	Op code	Mnemonic	Operand type	Bytes required	Op code	Mnemonic	Operand type	Bytes required	Op code	Mnemonic	Operand type	Bytes required
00	HALT		1	20	FORWARD	RC*	2	40	JUMP	R	2	60	SET	R,R	3
01	ZERO		1	21	REVERSE	RC	2	41	JUMPEQ	R	2	61	ADD	R,R	3
02				22	RIGHT	RC	2	42	JUMPNE	R	2	62	SUB	R,R	3
03	FAST		1	23	LEFT	RC	2	43				63	MUL	R,R	3
04	MEDIUM		1	24	RTURN	RC	2	44	SBROUTNE	R	2	64	DIV	R,R	3
05	SLOW		1	25	LTURN	RC	2	45	ARRAY	R	2	65			
06				26				46				66			
07	INCHES		1	27				47				67	COMPARE	R,R	3
08	FEET		1	28				48				68	GET	R,R	3
09	METERS		1	29				49				69	PUT	R,R	3
1A	MILLISEC		1	2A				4A				6A			
0B	SECONDS		1	2B				4B				6B			
0C	HOURS		1	2C				4C				6C			
0D				2D				5D	INCRMENT	R	2	6D			
0E	REPEAT		1	2E				4E	DECRMENT	R	2	6E			
0F	RETURN		1	2F				4F	CLEAR	R	2	6F			
10	FORWARD	R*	2	30	FORWARD	N*	3	50	JUMP	N	3	70	SET	N,R	4
11	REVERSE	R	2	31	REVERSE	N	3	51	JUMPEQ	N	3	71	ADD	N,R	4
12	RIGHT	R	2	32	RIGHT	N	3	52	JUMPNE	N	3	72	SUB	N,R	4
13	LEFT	R	2	33	LEFT	N	3	53				73	MUL	N,R	4
14	RTURN	R	2	34	RTURN	N	3	54	SBROUTNE	N	3	74	DIV	N,R	4
15	LTURN	R	2	35	LTURN	N	3	55	ARRAY	N	3	75			
16				36				56				76			
17	BEEP	R	2	37	BEEP	N	3	57				77	COMPARE	N,R	4
18	DISPLAY	R	2	38	DISPLAY	N	3	58				78	GET	N,R	4
19	HESITATE	R	2	39	HESITATE	N	3	59				79	PUT	N,R	4
1A				3A	CALL	N	3	5A				7A			
1B	TEST	R	2	3B				5B				7B			
1C	ENTER	R	3	3C				5C				7C			
1D	RANDOM	R	2	3D				5D				7D			
1E	SCAN	R	2	3E				5E				7E			
1F	PROXIMITY	R	2	3F				5F				7F			

Table 2: *Pseudoprocessor instruction set (*R = Register, N = Number, RC = Register Count).*

SCOTCH DISKETTES at 50% off!

ALL 3-M MEDIA AVAILABLE at 50% or MORE OFF LIST PRICE

Why settle for inferior substitutes when you can now get the very best media at unprecedented savings. Be it 8" or 5¼", hard- or soft-sectored: We are an **Authorized 3-M Distributor** and can ship it immediately.

Examples:

Part #		List	OUR PRICE (per diskette, in box of 10, individual envelopes)
744D	5¼" SSDD (hard/soft sector)	4.20	$2.04
745D	5¼" DSDD (hard/soft sector)	6.00	2.94
741D	8" SSDD (hard/soft sector)	5.55	2.75
743D	8" DSDD (hard/soft sector)	7.10	3.39

(even greater savings on diskettes in bulk pack—call for information)

Minimum order: 10 boxes. For orders of 1000 or more deduct 2%, for 5000 or more deduct 5%, for 10,000 or more deduct 7.5%. Call for savings on larger orders.

3M ®
Authorized Distributor

BHRT INC. P.O. Box 3791, Riverside, CA 92519

ORDERS & INFORMATION:
Mo.-Fri. 9:00 AM-5:30 PM PST, Sat. 12:00 AM-5:00 PM PST
1-(800) 845-8866 CA, AK, HI call (714) 783-1363

BANK REFERENCE: BARCLAYS BANK OF CALIFORNIA (213) 892-7244
AMERICAN EXPRESS, VISA, MASTERCHARGE
APO, FPO, INTERNATIONAL ORDERS ACCEPTED

Terms: Cashier's check, Visa, MasterCharge, American Express (add 4% for credit cards). Purchase orders from approved organizations on 30 day net at 2% surcharge. COD at 10% surcharge. Shipping charges extra.

Circle 506 on inquiry card.

BEST COPY AVAILABLE

devices and place the result into the specified register.

The next group of instructions, 20 through 2F hexadecimal, cause their respective motions until the proximity register changes; then motion ceases. The elapsed distance units are then recorded in the given register.

Instructions 30 through 3F hexadecimal are 3-byte commands that use an immediate 16-bit constant for the operand. For example, BEEP 7 (37 0007) signals the robot to beep seven times. In other respects, this group mirrors the group 10 hexadecimal instructions, except for the CALL instruction; CALL is a "hook" to an assembly-language subroutine, and its operand is a 16-bit absolute address.

Five instructions work with the processor registers. JUMP A takes the signed 16-bit value in register A and adds it to the program counter. JUMPEQ and JUMPNE are conditional jumps that test the condition code register. SBROUTNE is a simple subroutine call to a section of the program, and it terminates with a RETURN instruction. Each processor register instruction uses an operand that is a relative offset to the program counter so that the resulting code is position-independent.

The ARRAY instruction declares the array being referenced by adding its operand to the program counter and placing the resulting address in the array register. The INCRMENT, DECRMENT, and CLEAR instructions are added for convenience.

GET and PUT are used with the ARRAY instruction. After an array has been declared, a GET 5,B (70 0005 02) reads the fifth 16-bit element from the array and stores it in register B. The PUT A,B instruction (69 01 02) uses the number in register A as the offset into the array and places the number found there in register B.

Finally, the register-manipulation instructions use a constant or a register to perform an operation on another register. For instance, ADD B,C (60 02 03) means "add register B to register C." COMPARE compares a constant or a register to another register and then sets the appropriate condition code.

Some Examples

Armed with these instructions, let's try them on a simple program. Suppose we want our simple robot to search for a source of light in the room, determine how far away it is, and then report back to us. The resulting program might look something like listing 2. In this program, the robot turns to the right until it faces a source of light. Then it moves forward until it senses or bumps into the source of light, whereupon it moves back to its starting position, displays the distance, and stops. This program requires only 17 bytes through the keypad.

Of course, that is assuming the light sensor is mounted on the front of the robot and the path to the light source is unobstructed. If the program has to take obstructions into account, a triangulation scheme could be worked into it (but that, as they say, is left "as an exercise for the

★ ★ ★ FEATURING 8 AND 16 BIT SYSTEMS ★ ★ ★

EPSON QX VALDOCS: Extremely user friendly! See review Sept. BYTE. HX 80: Notebook-sized battery operated Z 80 computer, up to 256 K RAM, built-in hard copy, LCD scrollable screen.

MASTER MAX: S-100 system, Z-80, INTERCONTINENTAL CPZ48000 single card computer with four channels of DMA, dual 8" double density drives, CP/M .... $2,540.
Options: double sided drives, Winchester, TURBODOS, 2 user, 220v/50hz.

IMS 8X MULTIUSER SYSTEMS: Z-80, S-100. Each user has own Z-80, 64K RAM, 2 I/O. TURBODOS multiuser CP/M compatible operating system cuts link/edit time in half. Z-80 code. Interrupt driven. 8088 upgrade w/256K RAM has been announced.

TARBELL: Empire systems, Z-80, S-100.
CROMEMCO: C-10 personal computer w/software package .... $1,695.

8088: COLUMBIA DATA: IBM-PC look alike, multiuser option.

8086 S-100 SYSTEMS:
LOMAS: with MS-DOS or CP/M-86. Winchester option.
SEATTLE: with simultaneous 8" and 5" drives. Will accept IBM/PC software.

DUAL PROCESSOR SYSTEMS:
GODBOUT 816 A,B,C: 8085/8088. MP/M 816 allows simultaneous operation of both processors.
CROMEMCO DPU: 68000 and Z 80. CROMIX operating system.

MAX BOX 8" DRIVE SUBSYSTEMS w/QUMES, SHUGARTS, MITSUBISHI, NEC.

PRINTERS (dot matrix and LQ):
EPSON, NEC, QUME, C.ITOH, IDS, FLORIDA DATA, TELETYPE.

TERMINALS: WYSE, HAZELTINE, IBM 3101, TELEVIDEO.
Voice recognition board for TELEVIDEO 950.

PER SCI 277/299 DRIVES.

MODEMS: U.S. ROBOTICS 1200/300 DC HAYES compatible .... $525.

S-100 MAINFRAMES:
PARADYNAMICS, ECT, some TEI 12 slot still in stock.

GRAPHICS:
MICROANGELO GRAPHICS.
MIRAGE: new from SCION.
AUTO-CAD Interactive graphics software; for engineers, architects, designers.
HOUSTON INSTRUMENTS PLOTTERS, DMP-29 .... $1,775.

IBM PC ACCESSORIES: Extensive line including QUADRAM, SEATTLE, 8080/8086 EMULATOR (software).

IBM 3270: compatible equipment from Teletype Corp. Fast delivery! Cost effective!

We have an extensive product line including systems, peripherals, software, boards, drives, consulting services. Write or call for detailed specifications. We have knowledgeable technical staff.

WE EXPORT Overseas Callers: Phone (212) 448-6298
TWX 710 588 2844 or Cable: OWENSASSOC

JOHN D. OWENS Associates, Inc.
12 Schubert Street, Staten Island, New York 10305
(212) 448-6253 (212) 448-2913 (212) 448-6298

BEST COPY AVAILABLE

Listing 2: *Sample program written in ROTERP. The robot is commanded to scan the room for a light source, determine its distance, and display the information.*

```
0000    12    0001    START    RIGHT            1
0003    2B    01               SCAN             A
0005    2E    01               TEST             A
0007    42    FFF8             JUMPEQ           START
0010    30    02               COUNT FORWARD    B
0012    21    02               REVERSE          B
0014    28    02               DISPLAY          B
0016    00                     HALT
```

reader"). Imagine how much more difficult it would be in assembly language!

Let's try programming the robot to survey a room. An array can be declared to store distances and positions, data that can be acquired during the robot's trips back and forth throughout the room until it is familiar with its surroundings.

Even this simple robot can be taught to wake you when the sun comes up. Just give it explicit instructions to go to an eastern bedroom window and then let the program loop patiently until light appears. At dawn the robot will retrace its steps to your bed and beep insistently.

The Implementation

The implementation of ROTERP is easier than I would have expected. This version fits into a single 2K-byte 2716 EPROM (erasable programmable read-only memory). It is also fairly memory-efficient: the memory required for the registers, temporary locations, and a 10-deep nested call stack is only 128 bytes, leaving the rest of available memory for program storage. (I used a Motorola MEK 6800 D2 microprocessor kit to arrive at these figures.)

Figure 2 shows the modules that need to be written to form the complete interpreter. The first module allocates some memory for all of the 16-bit processor registers, some temporary storage for use by the interpreter, and enough memory for a subroutine call stack, which must accommodate as many nested calls as might be reasonably needed. You can use 1 byte to allocate as many as 256 general-purpose registers. The tradeoff is simply between memory used by ROTERP and the amount of memory left for program storage.

The initialization module is entered after the program is placed into memory. First it places the fixed starting address of the program into the program counter, sets up any interrupt-service-routine vectors that are needed, and initializes any device that needs it. Next it calls a subroutine that clears all of the general-purpose registers and places some default values into the special-purpose reg-

Martin Marietta Aerospace

Data Processing Opportunities

Martin Marietta Aerospace, NASA's prime Contractor for the Space Shuttle External Tank has immediate openings for Data Processing professionals. Because we actually manufacture the external tank, you'll get to see the actual results of your efforts.

COMPUTER PROGRAMMER/ ANALYSTS

Immediate opportunities exist for individuals experienced in:

- **UNIVAC 1100**
  ASC11 COBOL
  DMS 1100
  DDL, SDDL, DMU
  DML, QLP
  DPS 1100, TIP
  D/B Editor

- **APPLICATION EXPERIENCE**
  Shop floor control, Scheduling, Manufacturing, Inventory, Purchasing, Configuration Management, Quality Engineering.

- **DATA BASE OPENINGS**
  Analyst, Design, Administrators with above hardware, software and applications experience.

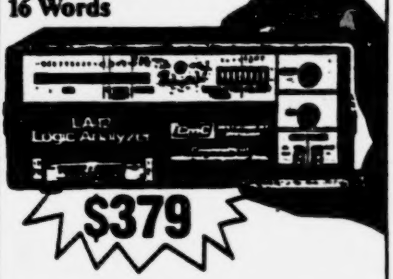
These opportunities exist at our Michoud Assembly Facility located in suburban East New Orleans.

Qualified candidates interested in learning more about these opportunities at Martin Marietta should forward resumes, including salary history to Martin Marietta Aerospace, Denver Glazier, BYTE-383, P.O. Box 29304, New Orleans, Louisiana 70189. We are an equal opportunity employer, m/f/h.

MARTIN MARIETTA

Everybody's Logic Analyzer

12 Channels
16 Words



A logic probe and oscilloscope are no longer adequate for analysis in today's digital world. For testing or debugging microcomputer or other digital logic circuits you need a real logic analyzer.

The LA-12 captures, stores and displays TTL and LSTTL digital data so that the instantaneous meaning of the data stream (e.g. data value, ASCII code, address) can be understood and analyzed long after the actual events have passed.

■ Easy to Use ■ 10 MHz ■ Clock Qualifier ■ Trigger input ■ 3 Trigger Qualifiers ■ Built-in LED Display — No oscilloscope needed ■ Compact ■ Expandable ■ Low Cost

30 day trial
Purchase an LA-12, use it, and if you are not completely satisfied, return it within 30 days and receive a full refund.

Free Offer
If you order within 45 days, *and mention this magazine*, you will receive a $49.95 input cable free with each LA-12 ordered.

Save $28.95
In addition, if you enclose payment with your order you can deduct 5% and we will pay shipping charges.

All prices are in US dollars for 120VAC.

To order in the Continental US call
TOLL FREE
1-(800) 547-5995 EXT. 195

Connecticut microComputer, Inc.
36 Del Mar Drive, Brookfield, CT 06804
(203) 775-4595 TWX: 710-456-0052

Q	Description	Price	Total
	Logic Analyzer	$379.00	
	Input Cable	49.95	
	20 Color-coded microclips	44.95	
Connecticut residents add 7½% sales tax			
	Shipping & Handling	$10.00	
	Total		

☐ Company purchase order enclosed (Rated Firms only)
☐ Check ☐ VISA ☐ MasterCard
Acct No. ____________
Signature ____________ Exp. Date ______
Name (Print) ____________
Address ____________
____
City ____________
State ____________ Zip ______
Dealer inquiries invited

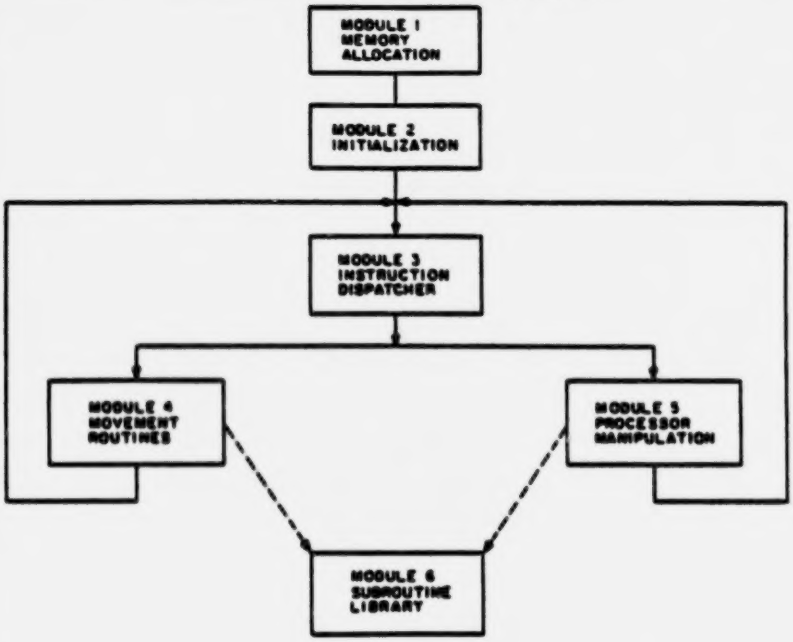


Figure 2: *Block diagram of ROTERP modules.*

isters. ROTERP uses SECONDS, FEET, and SLOW as defaults. This "clear and default" function is expressed as a subroutine because it is also called by the ZERO instruction.

Execution then continues into the instruction dispatch module, which contains a jump table used to pass control to the routine that executes the current instruction. If any invalid op code is executed, control is transferred to a routine that displays an error code and jumps the SBC monitor. This jump table is 256 bytes long because all op codes greater than 7F are currently treated as invalid instructions. If you want all 256 possible instructions, the jump table size would increase to 512 bytes.

The next section, which contains all the routines that carry out each instruction, is divided into two modules: one for execution of movement instructions and one for register-manipulation instructions. The instruction routines in the movement module are very dependent on the hardware scheme that controls the direction and speed of the motors. The motors can be interrupt-driven or controlled with timing loops in the manner of stepping motors and simple relay-operated DC motors.

All of the movement routines have many functions in common. First, they get the operands, translate them from a register designation to a working number as needed, and assign a temporary location to each working number. A scaling routine then translates the number of distance or time units into an absolute number expected by the motor handler. The op code itself serves as a direction indicator.

Next, the motor handler is called to perform the motion. And the last function every instruction performs is to add the length of the instruction to the program counter so that it points to the next instruction to be executed. Control is then passed back to the instruction dispatcher.

Because many of the instructions must perform common functions, the functions are expressed as subroutines and placed in the last module, the subroutine library. The subroutine library also includes routines that do common register operations like 16-bit addition, subtraction, multiplication, and division. The library is designed not only to save duplication of code but to permit the easy addition of new instructions.

To add a new instruction, you

BEST COPY AVAILABLE

simply place the new entry in the jump table and call the appropriate "fetch operand" routines. If the instruction is in the register-manipulation group, you can call many of the existing subroutines in the library to reduce the amount of new code that needs to be written. (If the instruction involves the use of an entirely new device, you must write a handler for that device, as you would in any case.) Finally, an existing routine increments the program counter and jumps back to the instruction dispatcher.

Extensions

If you find yourself using a particular sequence of instructions repeatedly, a single new instruction will make your program both shorter and easier to write. And the processor is not limited to expanding new instructions: a new device may be complex enough to warrant the use of a new special-purpose register. If you expect to use many different arrays, adding new array registers is only a matter of allocating space for them and assigning new op codes that call the same subroutines that the present ARRAY instruction uses.

The simple architecture and instruction set presented here should not be considered complete or taken for the optimum design of a pseudoprocessor for robots. Each design should be tailored individually to the kinds of experiments that you expect to do. For example, if you want a link to a host computer, the instruction dispatch module can be modified to receive instructions from a serial-line interface instead of reading them from memory. Two other options exist: the processor can be oriented around 32-bit registers if more precision is desired, or it can use 8-bit registers if less precision is needed. Expressing movement routines in terms of relative polar or Cartesian coordinates might be more desirable.

For maximum flexibility, one interpreter could include all of these features. If one byte does not sufficiently represent all of the different instructions, a 16-bit op code could be used. Better yet, a single instruction that changes jump tables would allow you to switch between whole instruction sets. The ultimate step forward, though, would be a translator that would produce running code from the instruction mnemonics. This translator would resolve symbolic references and might be the precursor of even higher-level "macro" statements of behavior. New language constructs, such as those found in LISP, might be implemented for ease of integrating the work being done in other areas of artificial intelligence.

Summary

ROTERP illustrates how a pseudoprocessor can control the increasing number of devices that can be incorporated into an experimental robot. With careful planning, you will be able to improve and expand the interpreter without difficulty. By tailoring the robot's processor to the kind of behavior it will perform, the robot becomes a much easier tool to work with.■

ALL 1/3 off SOFTWARE

Software Catalog. FREE. Get the lowest price on *every* software program you need. 1/3 off list price *guaranteed*. And look what else you get from ITM.

Unlimited Consultation! ITM's consultants work with an amazing database indexed with over 300 software selection criteria — plus thousands of in-depth product reviews. You'll get the programs that are right for you. Quickly. Easily. It's the most sophisticated consultation service in the industry. Call toll free!

Over 2,000 programs to choose from. Hundreds added every month. All categories. All formats.

See any program demonstrated. Order any non-entertainment program for *a risk-free, thirty-day trial*. Full refund if not completely satisfied.

No minimum order. Buy just one program if you like. Quick shipments.

All this, and more, is yours for an annual fee of $100. Call ITM now. Or, mail the coupon today and receive our FREE Software Catalog. Save hundreds—even thousands—of dollars on all of your software purchases with ITM.

American Express, Visa, or MasterCard honored.
Call toll-free today.
(800) 334-3404
In California (415) 284-7540

Software Catalog. FREE. Byte 383

☐ YES. Send complete information for my review. Please include a free copy of ITM's Software Catalog.

NAME __________
COMPANY NAME __________
ADDRESS __________
CITY/STATE/ZIP __________
PHONE __________

☐ Individual ☐ Dealer ☐ Consultant ☐ Company

ITM
Software Division
We make software buying simple.

Attn: Stevan Cloudtree
936 Dewing Ave., Suite E
Lafayette, CA 94549-4292
(800) 334-3404 or
(415) 284-7540

Circle 204 on inquiry card.

C. The Future of Robotics:

The state-of-the-art of robotics is changing quite rapidly. Since their beginning in the 1950s, robots have gone through what is called the low-technology and the medium-technology levels of classification. Currently the state-of-the-art is in the high-technology level classification. The tasks which each of these levels of robots can perform were identified in LAP #2. The important point illustrated in LAP #2 was that each succeeding level of robot can perform all the tasks which the preceeding level could perform. In other words, the capability of robots is increasing in proportion with the technologies which support them!

The future of robotics is difficult to predict. However, several trends, features, and/or improvements are identifiable. From these we can draw some conclusions as to what might be expected for both individuals and for society as a whole.

Two short readings are provided in this section (C) of this LAP. The first, by Thomas Emerson, concerns the possible technological aspects of the future of robotics. The second, by Pete Carey, expresses some of the concerns of society caused by the advancements taking place in the field of robotics. You should read both of these in order to have some knowledge of what may lie ahead!

From these, and other, readings several implications are apparent.

1. The technolgical advancements being seen today are an extension of changes which have been taking place for decades. Change is inevitable!

2. Most any change at least appears to carry with it both desirable and undesirable effects. The individual's responsibility is to study these thoroughly!

3. An individual's ability to change is typically proportional to one's preparation for it. This requires an open mind and education!

4. Throughout history, technological changes have eliminated some jobs, changed others, and created new jobs. Therefore, the time when an individual could hold one unchanging job throughout a lifetime is gone. Education or training for continued employment must, therefore, be a life-long continuing process.

5. Because automation will be used in many fields such as manufacturing, construction, inspection, food/fuel/fiber processing, communications, etc., an individual's preparation for the future should include a background in the sciences and mathematics on which this automation depends. Science studies should include physics, electronics, and hydraulics/pneumatics. The mathematics studies should include algebra, geometry, and trigonometry if at all possible. Additionally, social studies are important toward achieving a better understanding of the human concerns and of automation in general.

A variety of journals or periodicals is available concerning the field of robotics. Reading such literature should help keep you abreast of the happenings in this field. Also, several organizations whose concern is robotics have been formed in recent years. Communication with these organizations may help you to greater understanding of robotics and related fields. These journals/periodicals and organizations are listed below.

Journals/Periodicals:

1. Robotics Today.
2. Robotics Age.
3. Electronics.
4. High Technology.
5. Science Digest.
6. Radio Electronics.

Organizations:

1. American Association for Artificial Intelligence (AAAI), 445 Burgess Drive, Menlo Park, California 94025. Phone (415) 328-3123.
   American Federation of Information Processing Societies (AFIPS), 11815 North Lynn Street, Suite 800, Arlington, Virginia 22209. Phone: (703) 558-3600.
2. American Institute of Industrial Engineers, Inc., 25 Technology Park/ Atlanta, Norcross, Georgia 30092.
   Association for Computing Machinery (ACM), 11 West 42nd Street, Third Floor, New York, New York 10036. Phone: (212) 869-7440.
3. International Institute for Robotics, Box 210708, Dallas, Texas 75211.
4. Robot Institute of American, P.O. Box 930, Dearborn, Michigan 48128. Phone: (313) 271-0778.
5. Robotics International (RI/SME), P.O. Box 930, One SME Drive, Dearborn, Michigan 48128. Phone: (313) 271-1500.

From: Emerson, Thomas A. Technological Advancements In Industrial Robotics, Unpublished Masters Thesis; Department of Industrial Technology, University of North Dakota; May, 1984; pp. 46-55.

CHAPTER V

FUTURE TECHNOLOGY OF INDUSTRIAL ROBOTS

Introduction

The future technology of industrial robots is now being explored in numerous laboratories around the country. The primary areas of improvement are centered on mechanical aspects, sensor development and interface, vision expansion and simplification, and creative programming through several new techniques.

Mechanical Developments

Regarding mechanical aspects, the most sought after improvement is that of multiple appendage coordination. This refers to the simultaneous operation of two or more manipulator arms by one controller. Two variations of this concept are under consideration. One would be to have a given controller operate the movements of two or more isolated robots which are placed so that their work envelopes overlap at some point or area. Another would be to have a controller operate the motions of a specially constructed robot having two or more manipulator arms. In each case, the main objective would be to increase production through an improved handling system. A robot or system using one of these methods would be able to perform a much greater assortment of tasks than could a single arm system. Most of these would be related to assembly where it is necessary to orientate the parts before joining them. Multiple appendages would further eliminate the need for excessive clamps and fixture jigs required for some assembly tasks. The most attractive feature of these systems to users, however, is their potential for significant cost savings. With the first method, only one controller would be needed thus eliminating the cost of each one not required. The second method takes it one step further and accomplishes the work of two robots and controllers with just one complete system. The associated software for these systems is rather expensive, but the cost savings still outweigh the additional expenditure. A final advantage would be the more efficient use of floor space in the facility, thus allowing for potential expansion in equipment and production.

Another mechanical aspect is that of energy conserving robot kinematics. This refers to the construction of the robot base and manipulator arm in relation to actuator movement, location, and force generation. Through the use of CAD techniques, an optimal design may be derived which would provide for maximum flexibility, torque, and speed while minimizing the input power required, undesired inertial reactions, and uneven stress loading factors. These improvements would result in a much more efficient and productive system with an extended life span. The last area of mechanical aspects is centered on the development of general purpose end effectors. The ones currently being used in industry were designed to perform only one or two primary tasks. This is somewhat self-defeating since the primary purpose of the industrial robot was to function as a multi-purpose manipulator. It is therefore imperative to the optimal utilization of robotics that a general purpose end effector be developed. Doing so would translate into substantial cost savings for users who would no longer have to design or purchase individual end effectors for each task they wish the robot to perform. Production related savings would also be derived from the time saved which was previously spent repeatedly changing end effectors.

Electromechanical Developments

In the area of sensor development, one type currently being researched clearly stands out as revolutionary. It is known as a 'direct fourier transformation sensor,' and is constructed of a piezoelectric substrate coated with cadmium sulfide. This combination results in a photo-conducting film which is sensitive to the various reactions between the high-frequency sound waves and the photoelectrons that are produced by a focused-image light source. The associated software with this system can then instruct the sensor to pick out specified recognition aspects of an object. Advantages of this system include a reduced amount of data processing and of memory storage needed to analyze an image (25). Advancements in relation to vision systems are by far the most widely researched area of robotic high technology. One of the most interesting of such developments under consideration is that of a liquid crystal lens. This device would be mounted on the video camera in place of the standard lens and electronically focused by the corresponding vision system in a manner resembling that of human vision. The primary advantage here is in the elimination of moving mechanical focusing hardware which is susceptible to wear and induced error.

Another vision method being researched is that of optical circular scanning. This system is an approach at simplifying the recognition of objects through the use of a two dimensional circular tracing path. As an example, suppose a series of T-hinges have been manufactured and are placed on a conveyor belt with other objects which will pass under the video camera of this system for recognition, classifying, and rerouting to proper storage. When the edge of the object is detected, the camera moves out at a given radius from this point and performs a circular scan. Upon doing this it will detect other edges of the object in a fixed pattern. This process is done at two different radius values and the resulting patterns are noted. These patterns are then compared to those which were previously entered into the system's memory during 'learning.' If a match is found, the object can be counted, classified, or rerouted as desired. The primary advantages of this system over that of a standard binary or gray scale system is that it requires much less data processing in order to perform a basic recognition function. Therefore, it is easier for this system to scan, identify, and act upon the given object in real time which could result in higher productivity and associated cost savings. It also, however, has a few short-comings inherent in its design. One is that the object must have a solid silhouette in order for positive edge detection to take place, a another is that it cannot be used for quality control tasks where a complete image of the object is necessary for evaluation purposes (26).

The epitome of industrial vision systems, though, would be one which could produce a three dimensional color image of the object. The color capability is currently not within reach, but several systems are making substantial progress in the area of three dimensional color processing. Most methods of this type use two or more video cameras in their search for a three dimensional image. The separate images are broken down and defined as outlines which are then compared and oriented with respect to their common forefront edges. This blending then results in a three dimensional image as would be seen from the centerline between the cameras. The receding axes, however, are not always true-to-life due to the spacing and orientation of the cameras. The obvious advantage lies in the controller now possessing a three dimensional image which can be analyzed in regard to recognition and assembly tasks. The most unique system of this type may be viewed in Figure 18. Instead of using two separate cameras, this system incorporates a set of

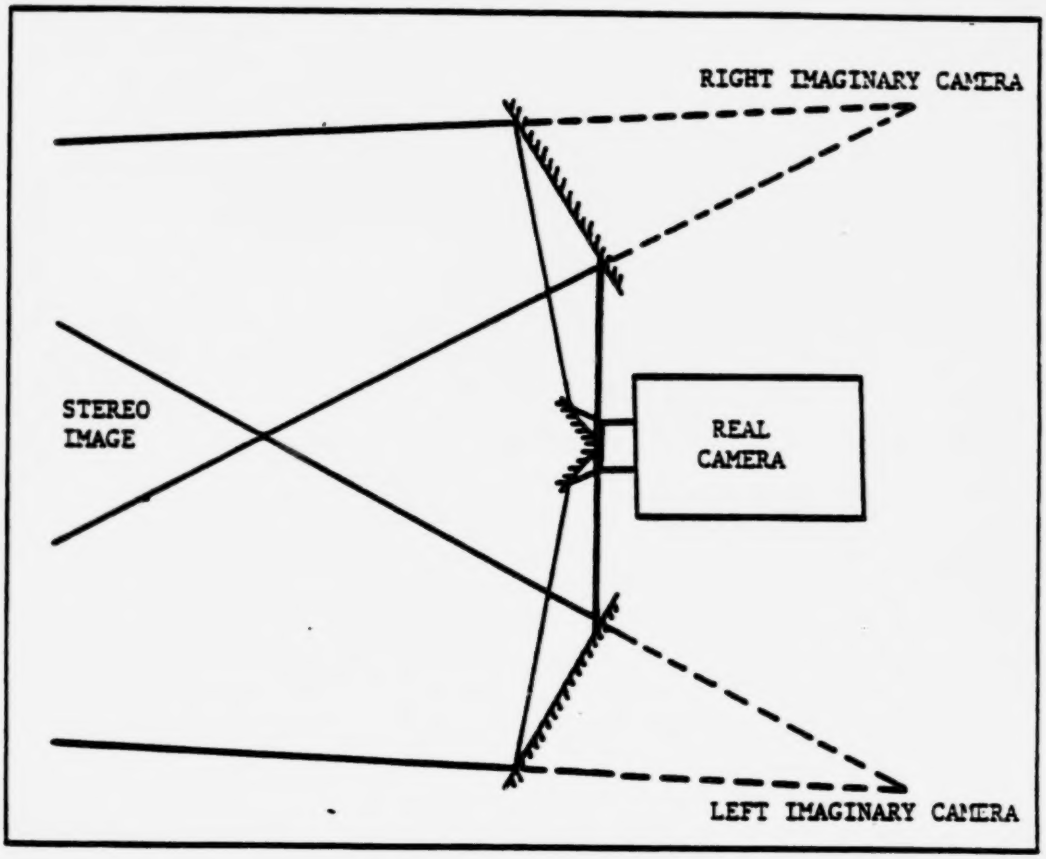


Fig. 18. Single Camera Three Dimensional Visual Imaging System.

SOURCE: Clifford Geschke, "A Robot Task Using Visual Tracking," Robotics Today 3 (Winter 1982-83):41, figure 3.

strategically located mirrors which result in a region emulating stereo vision. The object seen by the real camera is then the culmination of the two imaginary images. The advantage of this lies in the greatly reduced amount of vision data processing and integration that is required as compared with that of dual camera systems. A further advantage is the cost savings accrued in relation to the need for only one camera (27). A final note in regard to color image processing, though, is that some form of this capability is expected within the next one and one-half to two years.

Research is also being continued in the search for a universal off-line programming language. The need for such a language becomes more evident with each new system introduced to the market. It is getting to the point in some facilities that one department cannot utilize the information generated in the one next to it due to an incompatibility of the different languages used by the respective systems. This situation is analogous to the English/metric problem, only multiplied many times. Several interface devices are currently available to aid in this dilemma, but they are more of a temporary fix than a cure. Even once a common language is devised and adhered to by manufacturers, it will take awhile before an industry-wide dissemination is complete. Small facilities, however, would not benefit very much from using even a common language system of programming. In these plants, production runs are short which makes the extensive time needed for program development uneconomical. In light of this situation, and others similar to it, a new type of programming is being researched. It would allow a program to be developed through the use of a speech generator and a 'natural language' processor. This would be very beneficial where the tasks required of a robot are short-term and variable in nature. The system would have to be previously taught to recognize all of the expected voice input commands and the exact positions of the parts to be manipulated. The user could then instruct the robot by voice commands to perform a desired task with the parts. The following example would be typical of such a process.

```
(runvoiceio)

SPEECH INPUT:     "build a caster wheel"

RECOGNIZED:       BUILD A CASTER WHEEL

CONFIRMATION:     "yes"

SEARCHING PLAN LIST . . . NO PLANS FOUND FOR CASTER WHEEL

ENTER TRAINING MODE (YES/NO)?

SPEECH INPUT:     "yes"

TYPE GOAL NAME:  CASTER WHEEL

ENABLING VOICE TRAINING MODE

SPEECH INPUT:     "locate the wheel in the caster frame"

RECOGNIZED:       LOCATE THE WHEEL IN THE CASTER FRAME

CONFIRMATION:     "yes"

(operation is performed)

SPEECH INPUT:     "put the locating pin through the assembly"

RECOGNIZED:       PUT THE LOCATING PIN THROUGH THE ASSEMBLY

CONFIRMATION:     "yes"

SPECIFY:  DO YOU MEAN:  PUT THE HORIZONTAL LOCATING PIN THROUGH

                        THE ASSEMBLY

CONFIRMATION:     "yes"
```

```
(operation is performed)

SPEECH INPUT:    "insert the vertical locating pin in the assembly"

RECOGNIZED:      INSERT THE VERTICAL LOCATING PIN IN THE ASSEMBLY

CONFIRMATION:    "yes"

(operation is performed)


TRAINING CONCLUDED


BUILDING NEW PLANNING RULE:

GOAL:  CASTER WHEEL

PLAN:  1)  LOCATE WHEEL ASSEMBLY

       2)  PUT HORIZONTAL PIN ASSEMBLY

       3)  INSERT VERTICAL PIN ASSEMBLY


(runvoiceio)

SPEECH INPUT:    "build a caster wheel"

RECOGNIZED:      BUILD A CASTER WHEEL

CONFIRMATION:    "yes"

SEARCHING PLAN LIST . . . PLAN RETRIEVED

(assembly program is performed)

(exit)
```

This system would offer a great deal of flexibility to the robot, thus allowing it to be programmed in a minimal amount of time and handle a variety of ambiguous tasks (28).

Another area that is somewhat related to the performing of tasks which require a decision to be made while in motion is that of self-diagnostic fault tracing. This refers to the ability of a robot to conduct a test upon its mechanical functions with respect to their data accuracy. If a robot performed a task in which the end result did not correspond with the one initially programmed, it would conduct a series of tests in regard to its sensing and motion control devices in an attempt to locate the source of the deviation. Needless to say, this capability is strictly experimental at present.

The last major area of future robotics technology is that of AI or artificial intelligence. This concept involves the processing of symbols rather than only rigidly defined numeric values. The main drawback encompassing even the largest modern computers is that they do not possess any reasoning or logic capabilities. They are committed to endlessly processing numbers without actually learning anything. With an AI system, objects, concepts, and words are defined with respect to their characteristics which are then stored in a database. The recent advent of virtual memory machines has breathed new life into the development of AI systems, which had previously been put on a back burner in the '60s due to a lack of sufficient computer memory capabilities. An AI system can then search through this information and make comparisons, infer values, or logically arrive at an answer to an ambiguous question without having to operate within strict rules or exact instructions. This is sometimes referred to as DWIM or "Do What I Mean' processing. The most common system of this type is known as INTELLECT. AI reasoning ability could open a whole new era in robotics applications. No longer would every movement be dependent on an unyielding numeric value. The basic tasks and functions desired could be entered into the system and the appropriate actions would be self-generated or inferred with respect to its database and the continuously inputed sensing data (29). It is this capability that would transport the robot into the realm of a truly flexible manipulator.

BEST COPY AVAILABLE

From: Carey, Pete (Knight-Ridder Newspapers), "Steel-Collar Workers", Grand Forks Herald, Grand Forks, ND; February 26, 1984, pp. D1 and D4.

STEEL-COLLAR WORKERS

Led by hundreds of Silicon Valley engineers, the United States is rapidly moving toward a time when robots — clever, computerized and tireless machines — will take over millions of jobs and redesign America's workplace.

But as the revolution rushes onward, computer designers and other thinkers worry about the possible toll on society. Some say robots will give Americans more material wealth, free time and a chance for increased creativity.

But at what cost?

20.C.12

Robotics: Do gains for society offset the cons?

By Pete Carey
Knight-Ridder Newspapers

SAN JOSE, Calif. — Robot engineer James R. Young is worried.

He's designing a seeing robot that will replace a human assembly line at a company that makes electrical products. Soon he will begin automating the inspection for microscopic flaws in printed circuit boards, now a job done by humans. The job is going well, but . . . .

"An engineer, such as I am, we begin to feel people getting mad at us. Here we are, inventing all this stuff that's putting people out of work. . . . We've got to do something. But see, I don't know anything about that. I know about machine vision and that sort of thing," said Young, who works at SRI International in Menlo Park.

"You know, I don't even mind a little hysteria about this. Even that has some value. It's when a problem is ignored that it's really dangerous."

But whose problem is it?

The trouble is that there is little agreement about the extent of the problem, what to do about it or how to answer those questions.

"People come in and say, 'How do I automate this?' " said David Burwen, marketing manager with Integrated Automation, a company in Berkeley, that replaces human jobs with machine vision and other devices. "As part of our planning work, we ask what they are going to do with the people who get displaced. We will look for places in the company to shift them. We will provide training. We offer that, but very few companies are interested in it yet."

Burwen added: "The people who will lose jobs are the unskilled. It's a real challenge for the country. The education system is in bad shape right now. Government has to address that."

Economists are struggling with the issue, analyzing the possible effects of automation on the future well-being of society. Most are issuing pessimistic opinions, but they are groping without adequate data.

The people creating mechanical replacements for human workers debate it, too.

The issue of blue-collar job losses comes up "more often than you would think when you get a bunch of automation and robotics people together," according to David Grossman, an IBM robotics expert.

On almost any subject dealing with automation and robotics, some of the opinions of authoritative, distinguished figures in their fields are 180 degrees apart.

Contrast Grossman with Nils Nilsson of SRI International, for example. Grossman is a leading robotics specialist; Nilsson is one of the top people in the field of artificial intelligence. As Grossman sees it, there will be inevitable pains for some workers but overall gains for society. Nilsson, on the other hand, has a foreboding vision of a chasm of automation separating human workers and available work.

"There are some lessons to be derived from history," Grossman said. "The biggest one is the card loom, invented in 1804. The hand weavers were rather afraid the machine would take away their jobs. In 1811-15, a group called the Luddites went around smashing machinery.

"Were the Luddites right or were they wrong? It's very clear from a global point of view that automation was beneficial to humanity. On the other hand, all the hand weavers were thrown out of work, received small pensions and all had died out in 1850.

"The lesson is that there tends to be a small dislocation . . . and it is absolutely no consolation for the person who is affected to be told, 'You're a small dislocation in a good global picture.' "

But Nilsson says what's coming is a more radical dislocation, a basic shift in the definition of work.

"The main thing about artificial intelligence that contrasts with previous technologies for automation, which led to an expansion of the economy and more jobs," he said, is that "although artificial intelligence may do that also, the technology itself can perform some of these jobs.

"It's a little different thing we're involved with. Even if artificial intelligence creates more jobs, it may well be that artificial intelligence devices may fill some of those jobs."

Humans tamed and automated farming, only to open up millions of new jobs in manufacturing, Nilsson notes. "We are now shifting from manufacturing to service-sector jobs, and AI (artificial intelligence) is going to have a tremendous impact on service-sector jobs. The question is, where does the shift go from there?"

Peter Palecek, an SRI International robotics and automation analyst, expects robots to assume a variety of service-industry tasks that people think of as jobs with pay, fringe benefits and co-workers.

For example, he predicts, they will move into "the food industries on the assembly lines, making bread and cookies and all the other stuff, and moving materials.

"In the food services you can envision, 10 to 15 years from now, robots in McDonald's, replacing those teen-agers there."

Joe Avery, a manufacturing manager formerly at Hewlett-Packard Data Systems in Cupertino, is developing an automation plan for Activision. He has seen the impact of automation firsthand at Hewlett-Packard.

"So far, the effect has been that we haven't gone out and hired more people," he said. "But certainly, for those people who are jobless right now, there aren't as many entry-level positions as there once were."

Jobs are being created, Avery says, but they demand higher skills. Eventually, a point will be reached at which "there are more jobs being displaced than jobs created. I think we need to be concerned about it.

"The way we perceive work itself may have to change. We've adopted such a strong feeling about working," Avery said. "It's not supposed to be nice not to be working. The prospect of a lot of us not working, not being able to work, is something we have to address. We have to face up to it and start doing things about it."

Not everyone in the field agrees, however.

Jay M. Tenenbaum, director of artificial intelligence for Fairchild Camera & Instrument Corp. in Palo Alto, said he thinks the U.S. economy is a "self-correcting system" that will heal its wounds.

He says that, in response to automation, the workweek will be reduced but wage levels will be maintained. Companies still have to sell products to consumers, he reasons, so they will find a way to distribute some of their profits.

The specter of job losses caused by automation doesn't worry Bill Bottoms, president of Varian Associates' semiconductor equipment division in Palo Alto. Varian Semiconductor furnishes automated equipment to manufacturers trying to bring electronics factories back from overseas. The idea is to create automatic factories in the United States.

"Every time there has been a major change in the economy, there have been reasons to believe that we were going to have massive unemployment," Bottoms said. "But what's happened over time is that wages have risen and work hours have declined.

"As robots begin to take over . . . the workweek will go down and the standard of living will go up. . . . The next level of improved productivity is going to come from electronics and robotics."

If that is the case, everyone can stop worrying. But a number of economists are gathering data that make for gloomy reading.

A Carnegie-Mellon University team quotes Joseph Engelberger, the "father" of robotics, in a 1981 survey of the field. Engelberger said, "The prime issue in justifying a robot is labor displacement. . . . The key motivation is the saving of a labor cost by supplanting a human worker with a robot."

The Carnegie-Mellon team added, "all other evidence available strongly supports this view."

A study of technology and employment, published last year by SRI International and

BEST COPY AVAILABLE

the California Employment Development Department, predicts factory automation "will displace thousands of workers from some manufacturing jobs" in the 1980s.

The Rand Corp. released a study that predicts as little as 2 percent of the work force will be employed in manufacturing by the year 2000, down from the current level of about 20 percent.

At New York University, economist Faye Duchin studies technology and employment trends. There are "real reasons," she says, why it is not realistic to count on the economy growing indefinitely, creating new jobs for those that disappear.

"An increasing proportion of the adult population . . . may be excluded from the labor force of the future and without any sources of income," she said.

Duchin makes econometric models of technology and employment trends. She thinks she may have been too conservative in estimating the growth of robotics.

"What we found is not enormously dramatic — 200,000 jobs (replaced) by the year 2000. That's the first cut and reflects our conservatism. At the moment, we're (looking at) robots only in the jobs they'll be doing in the next five years, painting and welding — only the core occupations. I personally think this area is going to explode much faster than our scenarios have it."

Employment already has begun to level off in several industries, according to one economist. Richard Carlson, an SRI economist who co-authored the SRI study that warns of coming job losses, says most studies of employment trends mislead.

"You have a whole series of reports that say this is what happened in the 1970s, and then they draw a straight line and say this is what will happen in the 1980s," he said. "And we say it's just not true. There will be a very significant topping off (leveling off in employment). It's already starting to appear in finance, retail trade in general, in restaurants, some parts of middle management and lower-level computer programming."

The Club of Rome, an economics research group, warns in a recent study, "Micro-electronics and Society," that job loss is an almost certain consequence of automation. Some sectors of national economies will expand because of automation, but not enough to balance the loss in jobs, the research group reported.

"As a result of this development, many highly developed countries with already high rates of unemployment will not be able to reduce these rates and indeed will be in permanent danger of seeing unemployment rates rise even higher," the Club of Rome reports. "The persons affected most will be women."

Why women? Because the assembly jobs, the secretarial and clerical jobs most conveniently automated are predominantly held by women in many countries.

The Silicon Valley is no exception, according to Betsy A. Collard, director of employment services at the Resource Center for Women in Palo Alto.

"I'm not talking about things like retail, maids and janitors. I'm talking about office, industry and technical jobs — you don't see the shipping clerk jobs, the material handling jobs, the stockroom, the file clerks. The new technology is exciting," she said, "until you're the one faced with being laid off."

Those laid off might find new jobs, but they'd need new skills, according to the W.E. Upjohn Institute for Employment Research in Kalamazoo, Mich., which studied robotics for the state of Michigan. The institute predicted the robot industry would create up to 64,000 jobs making, installing and servicing robots by 1990, while up to 200,000 jobs would be eliminated by robots. The new jobs also would require much higher skills, the institute said.

"We found something we called the 'skill twist,' " said Alan Hunt, one of the two economists who did the Upjohn study. "The jobs eliminated and the jobs created are very different. The jobs to be eliminated are largely semi-skilled machine-operating jobs. The jobs created by robotics are much more highly scientific. That tells us we'll have a problem recycling workers."

Hunt views the situation as an accentuation of a 'long-term mismatch' extending over the past 20 years, between work and worker abilities in the United States. There are simply too many low-skilled people, he said.

But what is happening is normal, says Robert Ayres, who directed the Carnegie-Mellon study of robotics. "It's part of the 'gale of creative destruction,' " he said, using a phrase coined years ago by economist Joseph A. Schumpeter.

"Technological changes such as robotization are essential ingredients in the gale of creative destruction in which new industries come into being and old ones become rejuvenated or decline," he wrote in a new book on robotics.

Ayres says overall employment may or may not drop, but "there is a strong likelihood of displacement and unemployment among certain categories of industrial workers, especially in the older manufacturing centers."

And in his recent book, Ayres notes wryly that a 1964 congressional study by the "Ad Hoc Committee on the Triple Revolution" contends job loss from automation "would be so great that it would be necessary for the federal government to provide . . . costly income supports to a large fraction of the work force."

It didn't happen. Whole new industries opened up — the computer industry, for example.

Meanwhile, among the technologists busy making machines that replace humans, there is a sense that the task is a good one, a healthy one, even a liberating one.

Young, the worried SRI robot engineer, says he doesn't regret automating assembly line inspection jobs. It's a boring, inhuman job, he says. "I just hope those people get a different job, a better job."

And Nilsson, the AI expert, says he can't imagine why anyone would want to have "dreary, dull middle-management jobs" if they didn't have to.

"I've noticed that people regard the possibility of computers doing this sort of work as being somewhat scary," Nilsson said. "I understand why they have that point of view. Their jobs are going to be eliminated.

"But from a larger perspective, and to the extent that some of those jobs aren't all that interesting, why should it be scary? We're going to have these computer slaves working for us. It should be liberating."

APPENDIX A

RHINO ROBOTS INC.

XR SERIES
TEACH PENDANT

VERSION 2.0

RHINO ROBOTS INC.
TEACH PENDANT AND COMPUTER
XR SERIES SUPPORT HARDWARE
VERSION 2.0 (MARCH 21, 1984)

COPYRIGHT:
©1984 RHINO ROBOTS INC.

P.O. BOX 4010
CHAMPAIGN, IL 61820
(217) 352-8485

SPECIAL NOTE: 1.2 UPDATE

This TEACH PENDANT MANUAL introduces you to the exciting new second generation teach pendant for your Rhino XR series robotic arm. Keeping our current customers in mind, we have made nearly all of the new features available to them simply by purchasing two EPROMS.

The EPROMS are easily installed, following the instructions in the next section.

The new functions of the second generation teach pendant are the following:

1. The robot gripper acts as a pneumatic gripper with an OPEN or CLOSE command.
2. The robot can be stopped at any time during the RUN cycle by using the HOLD function.
3. The number of moves that can be stored in memory has been increased to 149.
4. The digital display indicates how many moves have been entered in LEARN mode or which move is being executed in RUN mode.
5. In STALL mode, the pendant provides feedback so that the robot can be brought out of a crash and the movement can be corrected, without loss of memory.
6. The ENTER key will not auto repeat or accept an entry of zero, which protects memory from erroneous entries.

7. Expanded visual display.

8. Improved EDIT functions.

9. Immediate escape from EDIT mode.

10. UPLOAD/DOWNLOAD is executed at 9600 BAUD.

11. Real time digital display of motor count.

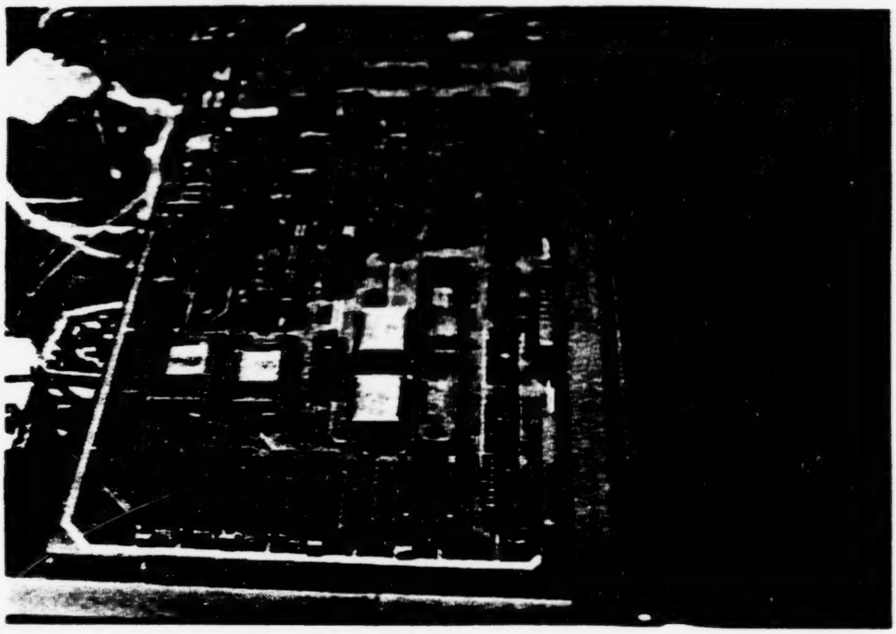
This is a brief summary of the main features. You will find many other improvements as you work with the system.

The only features that the TEACH PENDANT VERSION 1.2 does not have is the SHIFT/FWD and SHIFT/BACK (see page 34) functions which insert or delete moves in EDIT mode. This function could not be added without major hardware changes.

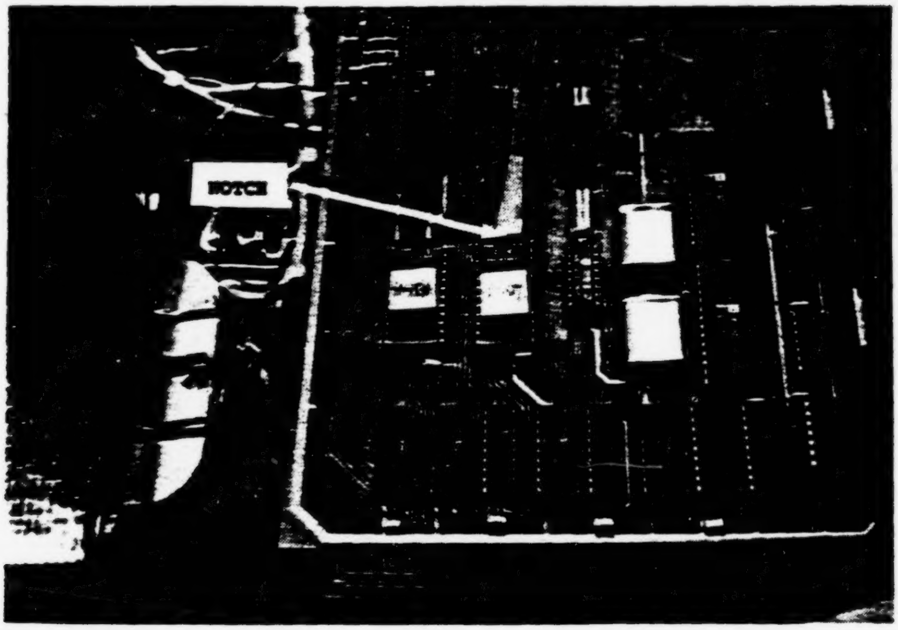
We hope you enjoy the new, expanded programming possibilities of the teach pendant system. Watch our USER REPORT for other exciting products.

CHANGING EPROMS ON THE TEACH PENDANT CIRCUIT BOARD

The EPROMs are on the teach pendant circuit board, upper board in the combination controller. The old (1.10) version EPROMs are labeled HIGH and LOW, as shown in Figure 1.

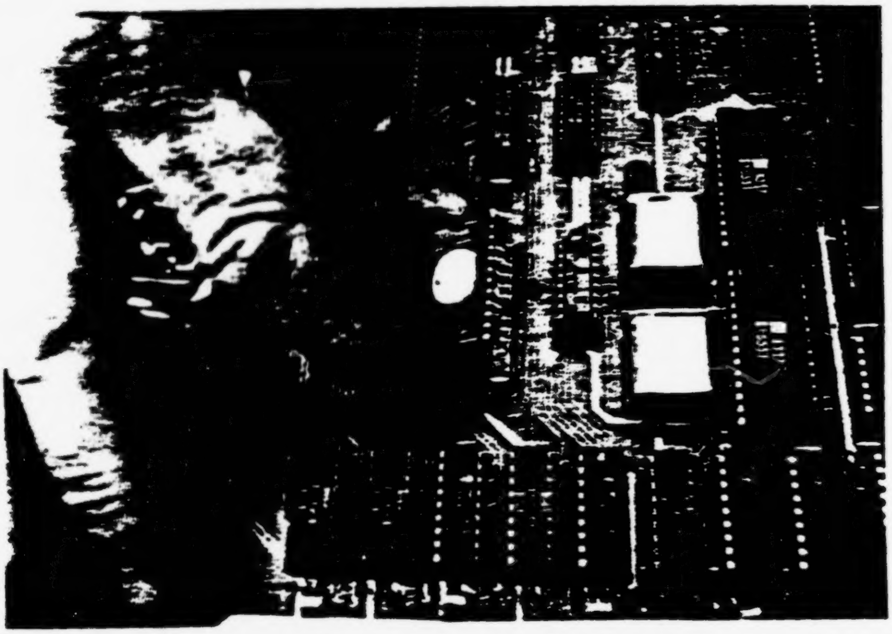


To change the EPROMs, gently pry the chips off the sockets, as shown in Figure 2. Note the notch on the chip. The updated version must be inserted with the notch in the same location.



Carefully bend the pins on the new EPROMs, as shown in Figure 3, to make them fit the sockets.

Insert the EPROMs in the sockets, beginning on one side, as shown in Figure 4. Check the location of the notch.



Press the new EPROMs in place when the fit is satisfactory, as shown in Figure 5. Make sure that all the pins are in the socket itself and none are bent underneath or outside the socket.

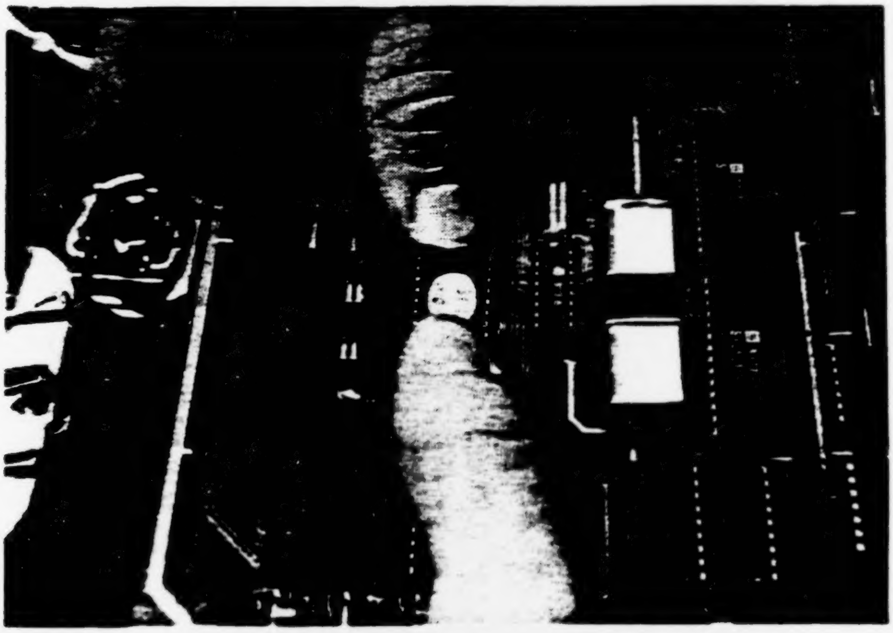
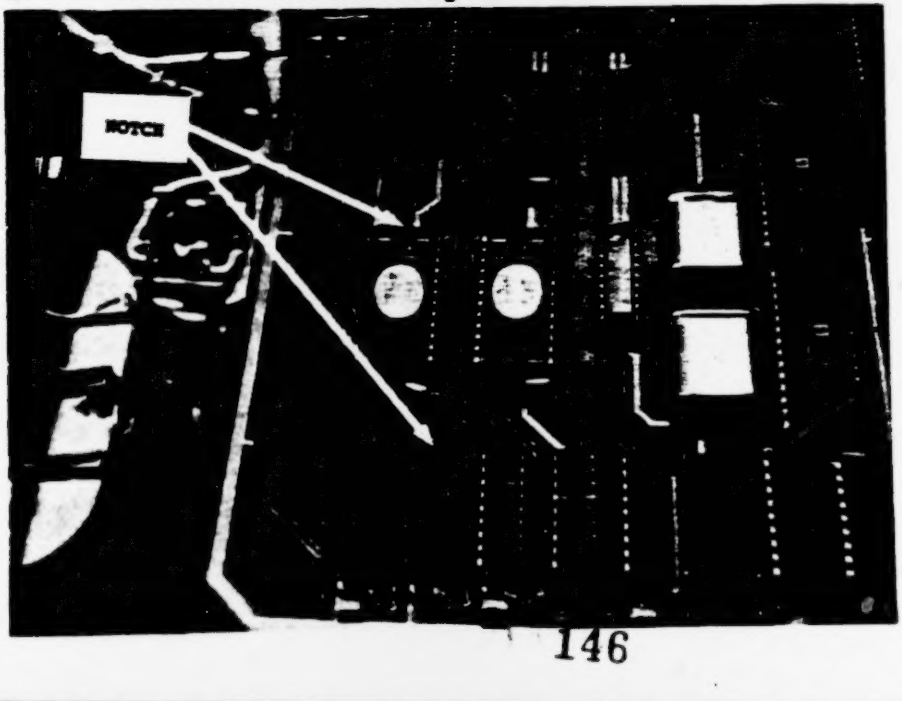


Figure 6 shows the teach pendant board with the new EPROMs labeled 1.15. Note that the notches must be in the same location as those of other chips on the board. The labels on the chips may be oriented differently.



CREDITS

The hardware for the teach pendant was designed by Carl A. Phillips, BS Electrical Engineering, University of Illinois.

The software/firmware in the teach pendant EPROM was designed and written by Foad Rekabi, BS Electrical Engineering, University of Illinois, and Thomas L. Redman, BA Computer Science, Southern Illinois University.

The following people's help is much appreciated:

1. Ellen Handler
2. Mark Carlson

We also would like to thank Mr. Harprit S. Sandhu who gave us the opportunity to introduce this product.

TABLE OF CONTENTS

1. INTRODUCTION ------------------------ 1

2. THE RHINO ROBOT TEACH PEANDANT SYSTEM ---- 2

3. SETTING UP -------------------------- 3

4. THE TEACH PENDANT KEYBOARD ------------ 5

5. USING THE TEACH PENDANT --------------- 7

5.1 INITIAL DISPLAYS ---------------------7

5.2 THE STRUCTURE OF THE TEACH PENDANT SYSTEM -- 7

6. PLAY MODE ----------------------------- 9

6.1 THE FUNCTION OF PLAY MODE -------------- 9

6.2 DISTANCE KEYS ------------------------ 10

7. LEARN MODE ---------------------------11

7.1 PRELIMINARY STEPS ---------------------11

7.2 SETTING THE "SOFT HOME" --------------- 12

7.3 TEACHING SINGLE MOTOR MOVES ------------ 13

7.4 RUNNING YOUR SINGLE MOTOR MOVE PROGRAM --- 14

7.5 TEACHING MULTI AXIS MOVES ---------------15

7.6 RUNNING MULTI AXIS MOVES --------------- 19

8. EDIT MODE -------------------------------20

8.1 FUNCTION OF THE EDIT MODE ---------------- 20

8.2 AN EXAMPLE -------------------------- 21

8.3 THE FWD AND BACK KEYS ------------------ 22

8.4 CHANGING MOTOR MOVES -------------------24

8.5 THE STEP KEY -------------------------- 25

9. UPLOADING AND DOWNLOADING A PROGRAM ------26

10. THE STALL ROUTINE ----------------------- 27

11. KEY FUNCTIONS REFERENCE GUIDE ------------- 28

12. SHIFT KEY FUNCTIONS REFERENCE GUIDE --------- 33

13. PENDANT DISPLAYS ------------------------35

14. WARRANTY AGREEMENT -------------------38

APPENDIX 1. Instructions for Upload/Download

APPENDIX 2. Software listings for Upload/Download

APPENDIX 3. Drawings and Schematics

APPENDIX 4. Troubleshooting

1. INTRODUCTION

Teach pendants are widely used for lead through programming of robots because they offer many advantages. All teach pendants are similar and use a readily understood pictoral language that can be mastered in a short time. Programming from a host computer requires familiarity with industry specific robotic languages such as VAL or AML, which contain many different commands and require considerable study before an operator can become familiar enough to control a robot with ease.

With a teach pendant, an operator can control a robot without knowledge of formal programming. Therefore, if an company wished to undertake a welding operation, it could ask an experienced welder to set up a move sequence even though the welder may not be able to program. A competent computer programmer may not know enough about welding to handle the assignment.

Another important group that make heavy use of teach pendants are maintenance and repair personnel. They need to conduct extensive tests of robot motion but do not need to create a complex move sequence from a host computer for which they would need formal programming skills. With a teach pendant, they can exercise all the axes and conduct trouble shooting operations.

Another advantage of a teach pendant is that the computer is transparent to the user. Since many people are still fearful of computers, it is helpful to introduce them to robot operation without discussing the computer that is part of the teach pendant system. The Rhino Robots teach pendant system includes a sophisticated microprocessor but the user does not need to be aware of this feature. A description of the electronic components of the teach pendant system can be found in Appendix 3 of this manual.

The Rhino teach pendant is ideal for rapid, hands-on introduction to the XR robotic system. Using the teach pendant, novices are able to begin operating the robot after just a few minutes of preliminary instruction.

2. THE RHINO ROBOTS TEACH PENDANT SYSTEM

The Rhino Robots teach pendant system consists of the following major components:

1. A 32 key, hand held pendant in a metal case.
2. A computer card that can be mounted in a separate controller or in a combination case together with the XR controller.
3. A software disk containing the Upload/Download program for saving and recalling move sequences.

   The Upload/Download program is described in detail in Appendix 1 and a software listing can be found in Appendix 2.

In addition to the above, you should also have the following items:

1. A jumper cable which should be short if you have the combination controller and long if you have separate controllers for the teach pendant and for the robot.
2. A flat ribbon cable that connects the hand held teach pendant to the teach pendant computer port.
3. A Rhino/Apple, Rhino/IBM or Rhino/TRS-80 cable that connects the teach pendant computer with the Apple host computer.
4. A user's manual.

3. SETTING UP

Unpack the teach pendant carefully and make sure that you have every item listed in the previous section.

Check that the power is off before hooking up the teach pendant system.

Set up the teach pendant system according to the configuration shown in Figure 1.

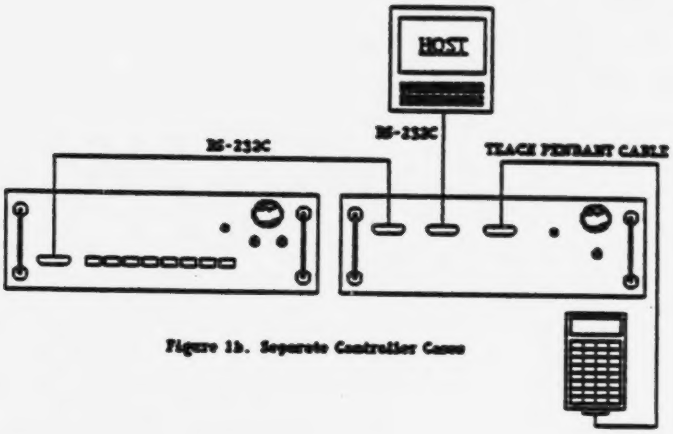

HOST
RS-232C
RS-232C
TEACH PENDANT CABLE

Figure 1b. Separate Controller Cases

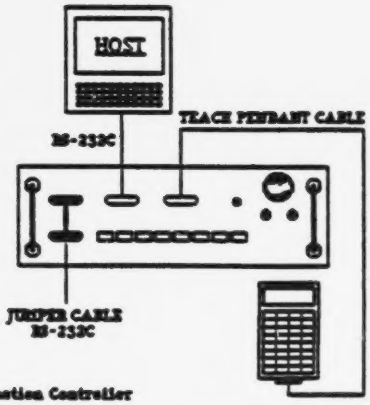

HOST
TEACH PENDANT CABLE
RS-232C
JUMPER CABLE
RS-232C

Figure 1a. Combination Controller

FIGURE 1. CONTROLLER CONFIGURATIONS

If you have a separate case for the teach pendant and for the robot controller, follow Figure 1a. You should have a long jumper cable to connect the RS 232C port of the teach pendant controller (left most port) with the RS 232C port of the robot controller (port directly to the left of the 8 motor ports.)

If you have a combination case (combo), follow figure 1b. You should have a short jumper cable to connect the RS 232C port on the teach pendant controller with the RS 232C port of the robot controller, directly below.

The teach pendant system is a stand alone device. This means that it does not require a host computer for robot operation. If you would like to save and recall programs that you have "taught" the robot with the teach pendant, you must plug the host computer into the center port of the controller, with the appropriate cable. Use your disk with the Upload/Download program in the host computer.

BEST COPY AVAILABLE

4. THE TEACH PENDANT KEYBOARD

The teach pendant display panel is shown in Figure 2.

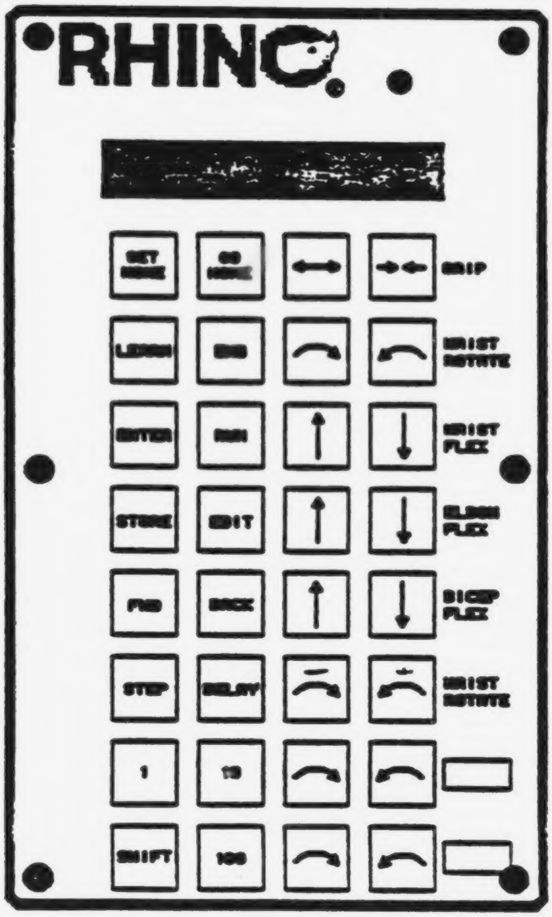

RHINO
SET HOME
GO HOME
GRIP
LEARN
END
WRIST ROTATE
ENTER
RUN
WRIST FLEX
STORE
EDIT
ELBOW FLEX
FWD
BACK
BICEP FLEX
STEP
DELAY
WRIST ROTATE
1
10
SHIFT
100

FIGURE 2. TEACH PENDANT KEYBOARD

A small, round light at top right is an indicator that power is on. Immediately below is an optical display that provides information on what is happening. As we describe use of the teach pendant, we will tell you what the different displays mean. You will find the information in the displays especially helpful when you are editing a move sequence. Most generally, the teach pendant display will show, from left to right, the mode, the motor currently being addressed, the direction of movement, and the number of steps on that motor.

The 32 keys on the keyboard can be divided into two parts. The two columns on the left (16 keys) include the function keys. The other two columns (16 keys) are the motor move keys.

The keyboard follow the structure of the XR robot, moving from the fingers to the waist. The keys correspond to the motors, designated A through H on the controller. Specifically,

Motor A corresponds to the finger gripper.
Motor B corresponds to the wrist rotation.
Motor C corresponds to the wrist flex.
Motor D corresponds to the elbow flex.
Motor E corresponds to the bicep flex.
Motor F corresponds to the waist rotation.
Motor G corresponds to the upper label - currently empty.
Motor H corresponds to the lower label - currently empty.

5. USING THE TEACH PENDANT

5.1 INITIAL DISPLAYS

As you power up, the word DELAY is shown on the display. After the display blanks out the gripper is reset to open. Then the word "SEt" appears which means that the system has found a temporary "Soft Home". As you move the robot and then press the GO HOME key, the system will return to that position. When you see the letter " P" at the left of the display, the system will have entered PLAY mode. This process is repeated at any time that the system is turned on or reset.

5.2 THE STRUCTURE OF THE TEACH PENDANT PROGRAM

The teach pendant has four main modes of operation, PLAY, LEARN, EDIT, and RUN. In PLAY you can try out various motor moves and get a feel for running a robot. Then you may teach the robot a designated sequence of moves in LEARN mode to carry out a specific task. You would use EDIT to ake minor modifications and fine tune the sequence you have taught. Finally, you cycle through the sequence in the RUN mode.

In addition, you can save a program using the STORE routine and recapture a saved program with the SHIFT/STORE command. Finally, the teach pendant program has a special STALL routine which helps you to recover from a situation where the system is blocked from carrying out a designated move sequence without losing the move sequence that the system was taught in the LEARN mode.

Figure 3 shows how the modes interact with each other. Notice that you always go back to PLAY when you switch from one mode to another.

Figure 4 shows how the four main modes interact with the STALL routine. Except when the system is in RUN mode, it can always go back to its prior mode from a STALL. If a STALL occurs while you are in RUN mode, you have to go into PLAY mode before you can go back to RUN.

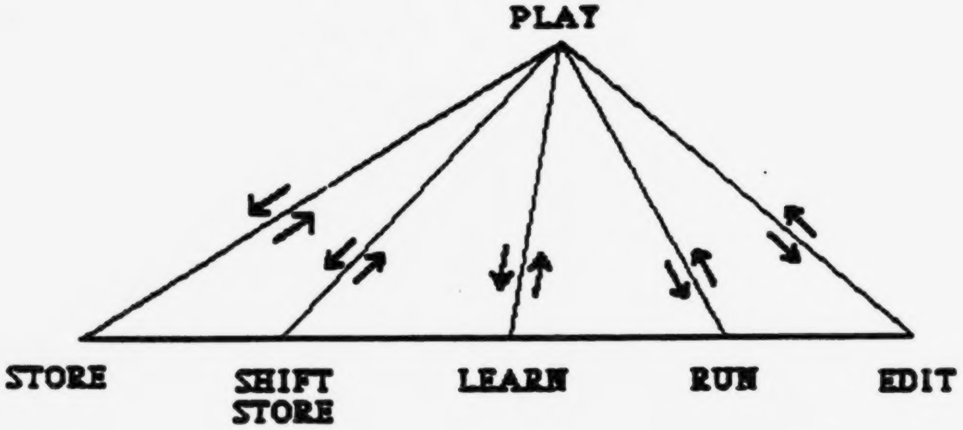

PLAY
STORE
SHIFT STORE
LEARN
RUN
EDIT

FIGURE 3. MODE STRUCTURE

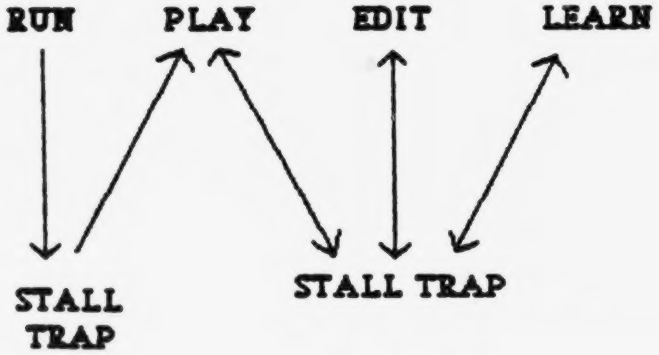

RUN
PLAY
EDIT
LEARN
STALL TRAP
STALL TRAP

FIGURE 4. MODE INTERACTION WITH STALL TRAP

6. PLAY MODE

6.1 THE FUNCTION OF PLAY MODE

After the DELAY disappears, the letter "P" will show, indicating that the system is now in PLAY mode. PLAY mode is very useful for practicing robot movements and positions. You can test the reach of the robot and find out how best to get to a designated location.

In PLAY, the motor move keys in the two right hand columns are active. By pressing any of these keys, you will be moving the corresponding motor on the robot.

The four motor move keys at the bottom, opposite the empty labels on the right, should be used only if you have motors in ports "G" and "H" of the robot controller. For instance, any of the motorized Rhino accessories such as the conveyor, carousel, slide base, or XY table, would be plugged into ports "G" or "H". Then, pressing the corresponding keys would move the accessories in the same way as the motors on the body of the XR robot.

CAUTION: Be careful to use motor move keys only when a motor has been plugged into its respective port. If a motor move key is pressed without a motor in that specific port, the system will lock up and you will lose your program.

If you keep your finger pressed on a motor move key, the AUTO REPEAT function will become active. The motor will move smoothly and continuously until you lift your finger.

As you press motor move keys, you will find the robot movement reflected on the keypad display. For instance, if you rotate the wrist counter clockwise 120 steps, the display will show PB - 0120. This means that the device is in PLAY mode and that the "B" motor has traveled 120 holes in the negative direction. The display always indicates the status of the last motor moved. The only exception is the end effector which has two settings, open or closed, to simulate a pneumatic gripper.

6.2 DISTANCE KEYS

The keys at the lower left of the pendant with the numbers 1, 10, and 100 are used to set the distance moved by a motor with each key press. The 10 key means that a motor will move 10 encoder counts for each press of a motor move key. This is the most usual setting and the default value.

Use the "1" key when you are fine tuning a move. For instance, you may want to use small increments together with the auto repeat feature to achieve exact placement of the robot gripper. The on/off or toggling gives the robot a slower, somewhat jerky action. It should be used any time that you are within two inches or so from your pick up or release point.

The "100" key is useful for doing long moves quickly. You might find it useful for swinging around the arm by moving the waist from one side of the robot to the other. You would not use the "100" key on the fingers of the robot. The "100" key is normally used with a conveyor or XY table where many moves are needed. This key can easily get you into a crash situation if you are not absolutely familiar with the product and should be used sparingly.

7. LEARN MODE

7.1 PRELIMINARY STEPS

In LEARN mode, the robot can be taught a set of moves that will be retained in memory. Before entering LEARN, two prior steps are necessary: CLEAR the memory, and send the robot to HARD HOME for its initial position.

To CLEAR the memory, do one of the following:

1. Push the RESET button on the teach pendant controller.
2. Push the SHIFT and END keys, either simultaneously or press SHIFT first and END immediately thereafter.

Before sending the robot to HARD HOME, set up the robot hand so that the SHIFT/GO HOME command will send the robot to a good HARD HOME on all axes. "Hard" home means "Hardware" home. It is used for initializing the robot. After the SHIFT/GO HOME command is given, the robot moves around and finds the exact center of the three interrupt switches on the "F", "E", and "D" motors. These provide an absolute position from which the robot can LEARN a move sequence. You must do a HARD HOME before you can enter LEARN mode. In case you forget, the display will remind you by telling you to"dO hArd".

Use the wrist flex keys to bring the hand into a vertical position and the wrist rotate keys to bring the fingers parallel to the arm frame. You may remember that this is the packing position. When the robot arrived, the fingers were in the parallel position, pushed down between the two large sprockets above the base. The fingers should be open about half way.

7.2 SETTING THE "SOFT HOME"

To establish a "Soft Home", or software home, move the robot into a desired configuration and press SET HOME. Thereafter, the robot will return to this position when the GO HOME key is pressed. The "Soft Home" will be retained until the power is cut off, or the RESET key is pressed. From now on, when we simply refer to "Home", we will always mean the "Soft Home". The "Soft Home" is concerned with all eight axes, not just with the three that "Hard Home" refers to.

The "Soft Home" increases robot efficiency. It should be set at a location that is convenient to the "pick up" position, about two inches above, in most cases. The "Soft Home" then becomes the actual starting point of the move sequence and the robot will not need to return all the way to "Hard Home" when it runs the move sequence.

The "Soft Home" position cannot be set unless the user has done a "Hard Home" first. If the user does not set a "Soft Home", the robot will treat the "Hard Home" as a "Soft Home" and return to the initial position when the GO HOME key is pressed. If you cannot remember whether you have set a "Soft Home" or not, simply move the robot a few steps and send it HOME. If it goes to "Hard Home", it is assuming that this is "Soft Home" as well. If the robot does not move when the GO HOME key is pressed, you can assume that it is already in its "Soft Home" position.

The robot needs to be sent to "Soft Home" before it will enter any other mode from PLAY. Therefore, if you find that you can't make the system enter another mode, check to see whether the robot is at "Soft Home".

7.3 TEACHING SINGLE MOTOR MOVES

Single motor move sequences are good for training and practice. Very quickly you will be developing multiple motor move sequences, which are more like the real world.

For a single motor move, you simply press the key that corresponds to the motor you wish to move until the robot has reached its destination. If you overshoot, press the key for the opposite direction until the robot is positioned correctly. Then press ENTER to record the move. Any back and forth motions that you may have made as you position this motor will not be reflected in the recorded move. For instance, if you turned clockwise 2000 steps on the waist, motor "F", and came back 430, your move would be registered as -1530.

After you ENTER the "F" motor move, your display changes. When you began, you had "L" on the left for LEARN and "0" on the right because no motor moves had been entered. After the waist rotation, your display will read

```
  L       F          -          1530
MODE    MOTOR    DIRECTION     #STEPS
```

Now when you record this move by pressing the ENTER key, the display will change to

```
  L                                1
MODE                            #MOVES
```

If this was the only movement you wished for your sequence, you would complete the cycle by pressing the GO HOME key, which sends the robot to the "Soft Home" position. If you did this, the display would change to

```
  L                                2
MODE                            #MOVES
```

Pressing the ENTER key always erases the step count display and updates the motor move count on the extreme right.

7.4 RUNNING YOUR SINGLE MOTOR MOVE PROGRAM

To complete the program, push the END key. The "L" on the left side of the display will disappear, signifying that the LEARN sequence has ended. The END key is always used to change modes. Now the "P" will appear, because the robot is again in PLAY mode.

Then you can push the RUN key and watch the robot carry out your instructions.

While the robot is in RUN mode, the display reflects its new condition. As before, the mode is at the left and now the move number is at right. During the first move in the sequence, the display will show

r	1
MODE	*MOVE

While the robot is returning HOME, which is the second move, the display will read

r	2
MODE	*MOVE

After you have pressed RUN, the robot cycles until you press RUN again or END. The second time you press the RUN key, the robot stops. The teach pendant display shows "HOLD" until you press the RUN key again to continue the cycle. This feature is useful for instructional presentation and for collision avoidance. If you press the END key, the robot completes its cycle before stopping. Thereafter, the display shows a "P" on the left to indicate that the system has reverted to the PLAY mode. After running a sequence of moves, you may go back into LEARN and add other moves to the end of the previous move sequence. Keeping this in mind, you may make a complex series of moves by simply stringing together LEARN sequences.

7.5 TEACHING MULTI AXIS MOVES

The essential difference between multi axis and single axis moves is in timing of the ENTER command. In single motor moves, you pressed ENTER after one axis had been moved to bring the arm to a given location. In the multi axis move, you press ENTER after you have moved any combination of axes to achieve a given point in space. Your preliminary steps are identical, i.e. do a "Hard Home", CLEAR memory, set a "Soft Home", and enter the LEARN mode.

When planning a multi axis move, keep in mind several important principles of robot movement.

PRINCIPLE #1. Start moving your robot from the waist up and move the fingers last. For instance, if you are planning to reach a point which requires waist ("F" motor) and bicep ("E" motor) movement, you start with the waist. As you move the motors, you will see the appropriate displays.

After you have moved the waist, your display might read

```
     L        F                        1370
   MODE     MOTOR       DIRECTION    #STEPS
```

There is no direction indicator if the move is in a positive direction (counterclockwise on the "F" motor).

Now switch to the bicep ("E" motor) without pressing the ENTER key. Notice that the display changes. Your display may show

```
   L        E              -       420
  MODE    MOTOR       DIRECTION   #STEPS
```

Your motor reading has changed from "F" to "E". The number of steps apply to the "E" motor only. You still have a 0 on the right because you have not recorded any moves in this sequence even though you have moved two axes.

Now, if you push the ENTER key, your display will change to

```
   L                                 1
  MODE                            #MOVES
```

You could have moved through all the other axes before recording the move by pressing the ENTER key. Remember that you can also readjust an axis before you record a move. Just press the key for the opposite direction on the same axis. Even after you have recorded the move, don't worry if your position is not exactly as you had wished. You can modify any move later with the EDIT capability, a very useful feature of the Rhino Robots teach pendant.

A further principle to remember is that you are moving the robot from one point to another, not a specific path. In order to avoid crashes, it is best to remember the points at which you ENTERED a move and try to estimate the path that the robot will take. After a while, you will become quite proficient but you may need patience at the beginning.

PRINCIPLE #2. Make the gripper OPEN/CLOSE a separate movement. Think about your own arm. If you wanted to pick up or put down an object, you would move your arm to the chosen location and then open or close your fingers. In order to appreciate this point, think about what would happen if you did not follow this principle. Your hand would open before your arm had moved to the place where you wanted the object placed, and whatever was in your hand would be dropped along the way. If you wanted to move your arm to a specific location to pick up an object, you would close your fingers on thin air, or on the wrong object, if you closed your fingers before your arm was properly positioned.

If this doesn't make sense, try an experiment. Ask another person to talk you through a move with your own arm while you are blindfolded. If the objective is to pick up an object from a desk top, you will find that your helper will direct you first to turn your body, then to lower your arm, moving it forward and back, and finally to open your fingers when you are correctly positioned.

PRINCIPLE #3. Each move should be in only one direction. Again, the best way to appreciate this principle is to think about what would happen if you failed to observe it. Suppose that you wanted to set an object down at the far side of a divider. You would raise the arm, move it over the divider, and lower it on the other side. If you did all this in one movement, the arm would take the shortest route to the end location and would crash full force into the divider that you were trying to avoid. As you may remember, the controller combines all the positive and negative motor moves on the same axis in the same movement. The algebraic sum makes the robot more efficient when you are practicing but it can cause problems unless you keep the process in mind. If you wish to make the robot move in two different directions, you must make a seperate entry in each direction.

PRINCIPLE #4. Watch how the robot moves from point to point to accustom yourself to its revolute coordinate system. The XR, like many industrial robots, moves in an arc from point to point. The arm moves from one exact location to another, but because of its coordinate system, it does not move in a straight line. Until you becasue accustomed to this facet and can accurately predict exactly how the robot will move, it is best to keep all obstacles out of harm's way.

PRINCIPLE #5. Avoid complicated, multi axis moves, especially in the beginning. It is better to ENTER often and to use many single axis moves. There is less likelihood of a "serious" crash (XR crashes are never really serious like those of the big robots) and it will be eaiser to EDIT a less complicated move sequence.

PRINCIPLE #6. Remember that when you tell the robot to GO HOME at the end of a move sequence, it will go to its "Soft Home" position in a single, multi axis move. If this is likely to cause problems, lead the robot to some intermediate locations, recorded with the ENTER key, before you give the GO HOME command. For example, if you set an object down, remember to move above the object after you release it. Then push the ENTER key. This will stop the XR from knocking down the object on its way home.

7.6 RUNNING MULTI AXIS MOVES

After you have finished the program, press the END key and then the RUN key. Remember that you can PAUSE in the cycle by pressing RUN a second time and then pressing RUN when you are ready to resume the cycle. If you press END during a run, the robot will stop at the end of the cycle and return to PLAY mode.

If you are planning a lengthy sequence, you may want to program it in sections. You might program a module, run it for testing, and possible editing, and then program another module. Under those circumstances, you would not want the robot to cycle continuously. After you had run the sequence once or twice, you would return to the LEARN mode and program some more.

Keep in mind that when the robot leaves the RUN mode, it automatically goes into PLAY. As you handle the teach pendant, you are very likely to touch some of the motor move keys which takes the robot away from its "Soft Home" position. Therefore, before you go back to LEARN mode, press GO HOME to make sure that the robot starts from the same "Soft Home" position where it had ended the prior move sequence.

Before beginning your second module, look at the teach pendant display. It should shown an "L" on the left for LEARN and a number other than "0" on the right. The number should be the same as the final count (including the return "Home") from your previous move module.

8. EDIT MODE

8.1 FUNCTION OF THE EDIT MODE

In EDIT mode, you can trouble shoot or modify a program that had been entered into memory. Using EDIT is similar to writing the original program but even easier.

The ability to edit a program is very important for efficient robot operation. In many industrial situations a program may be written off-line and loaded into a teach pendant. The programmer who is removed from the scene may not know exactly where the robot is to be located and where the vices and other fixtures will be placed. Such factors are very important for the proper operation of a robot and may require relatively minor modifications. The operator can bring the program to the plant floor and then make adjustments for location of fixtures.

Another common example could be a program that is working perfectly but changes in the pick up or delivery point require minor program changes. When that occurs, it is far more efficient to make changes in the step counts of specific motors than to re-write the entire program.

Of course, the EDIT mode will not work unless you have used LEARN mode to ENTER a program. If you have forgotten to use the ENTER key and push EDIT, the display will remind you by saying "NO DATA". You enter the EDIT mode from PLAY, the switching mode in the XR teach pendant system.

8.2. AN EXAMPLE

To give you an idea of how EDIT works, we have designed a simple example. Suppose you had entered the following program in LEARN mode on your teach pendant:

Move 1	F - 300	waist clockwise
Move 2	E 250	bicep out
	D 250	elbow down
Move 3	G 790	conveyor forward
Move 4	F 300	waist counterclockwise
	C - 250	wrist in
	B 500	fingers counterclockwise
Move 5	H 600	rotary table clockwise
Move 6	A close	
	B - 250	
	C 300	
	F 300	
Move 7	GO HOME	
	END	

Your display has now returned to PLAY and shows a "P".

8.3. USING THE FWD AND BACK KEYS

When you push EDIT you will see on the display:

```
        E                                   1
        MODE                              #MOVE
```

which means that the first move is being run. This display is very brief. After the move is complete, the display will read as follows:

```
        E         H            -        0000
        MODE      MOTOR     DIRECTION  #STEPS
```

This shows you what happened in the first move. If you look back at your program, you will see that the "H" motor was not moved before the first ENTER. In EDIT, you will see a display for every motor in every move. If a specific motor was not moved, it will show 0000 in the step section of the display.

Now press the FWD key and the display will go to the next motor and show

```
        E         G            -        0000
        MODE      MOTOR     DIRECTION  #STEPS
```

which proves that you did not move the "G" motor either before the first press of the ENTER key.

Press the FWD key again and the display will read

```
        E         F            -        0300
        MODE      MOTOR     DIRECTION  #STEPS
```

showing that you moved the "F" motor or waist 300 encoder holes or steps in the clockwise direction.

Press the FWD key again and you will see the displays for the other motors, "E", "D", "C", "B", and "A". For each motor, you will see a number except for the "A" motor which will show only OPEN or CLOSE which shows the condition of the fingers or end effector.

Now try the BACK key. You will see displays for motors "B", "C", "D", "E", "F", "G", and "H", moving in the opposite direction from the FWD key. To change the position of the "F" motor, you would have to press the BACK key until you found the "F" motor display and could change it. Remember, the FWD and BACK keys will only select a motor within a move. They do not move the system from one move to another.

8.4 CHANGING MOTOR MOVES

For changing the values, first make sure that the motor you are addressing is hooked up. This is especially important for the "G" and "H" motors which are usually reserved for accessories such as the conveyor, rotary table, slide base, or XY table. Because you can crash the system if you try to change "G" and "H" motor values when none are present, it may be wise to keep accessories in these ports or even to keep two empty motors plugged in.

You use the FWD and BACK keys to find the display for the motor where you want to make a change. In the beginning, until you become very familiar with the robotic system, it is advisable to change only motors that were moved, have non zero values, when you set up the move sequence in the LEARN mode. Beginners may find that the trajectory changes in unexpected ways especially when zero value moves were altered. That means that in the first move in the example above, you should change the count only for the waist, or "F" motor, but not the other motors.

Let us assume that you decided that you wished to change the F motor in the first move from -300 to -500. The F motor display on the first move will now read

```
  E        F              -         0500
MODE     MOTOR        DIRECTION   #STEPS
```

Once you have changed a value, you do not have to ENTER this change. Now step to the next move. The F motor in the second move will be adjusted to reflect the change you made. The new display will read

```
  E        F                        0200
MODE     MOTOR        DIRECTION   # STEPS
```

Adjusting the motor count helps the system to maintain the location you designated.

In the next section, we will show you how to step both forwards and backwards through the program. This is important because it allows you to back up and then step forward. In this manner, you may inspect a move sequence that you have modified. In a long sequence, this is quite helpful.

8.5 THE STEP KEY

When you are satisfied with the motor values in the first move of the sequence, press the STEP key to get to the next move. Right after you have pressed the key, you will see this display:

```
   E                                   2
 MODE                                #MOVE
```

This display disappears very quickly and after the robot has run through its second move, it will stop and the display will read

```
   E        H             -        0000
 MODE     MOTOR      DIRECTION   #STEPS
```

Use the FWD and BACK keys just as you did on the first move and change motor counts as needed. You can change the "E" or "D" motors within the second step in the sequence, but be careful if you alter the 0000 values.

When you wish to inspect the last move you edited, you press SHIFT/STEP. You may press this combination repeatedly to "Back Up" several steps and see the flow or a modified sequence. After using the SHIFT/STEP to "Back Up" moves, you would then use the STEP key to move forward through the next moves to decide if you have modified the sequence to your satisfaction. You can use the FWD and BACK keys within each step as before if you wish to modify or edit further.

Continue to use the STEP key until you have moved through the entire program. If you try to move beyond the last move, your display will show "LAST", meaning that the final move has been done. At that time, the only commands that the pendant will accept are the END command, which takes you out of the EDIT mode, or the SHIFT/STEP command which takes you back to preceding steps.

Another very useful feature of the EDIT function is that you can leave the EDIT mode without stepping through all the moves. This can be very handy if you are working on a long move sequence with many "Enters" and you do not want to step through the entire program.

9. UPLOADING AND DOWNLOADING A PROGRAM

After a program has been run successfully, you may want to save it on disk for future use. To do so, you must be in PLAY mode and then use the STORE key. Make sure that the system is properly connected to a host computer with the proper interface. Specific instructions for use with Apple II, IBM PC, and TRS-80 along with software listings can be found in Appendix 1 and 2.

XR users with other computers will need to program their own upload or download program using the following transmission protocol:

1. The STORE command sends a number indicating how many moves will be transmitted to the host computer. This is the number of moves in the buffer plus 1. The extra move is the distance between the "Hard Home" and "Soft Home" positions which is the first move to be transmitted.
2. Then it sends the move sequence.

The host computer must be programmed to receive these moves and store them in the proper variables in real time. The information is sent through the central serial port at 9600 BAUD, no parity, 8 data bits, and 1 stop bit. To slow down the pace of transmission, automatic short delays seperate the transmitted moves. As each move is transmitted, it is counted down at the right of the display. During uploading, the letter "U" will appear at the left of the teach pendant display.

To download a move sequence that was saved previously, press SHIFT/STORE in PLAY mode and make sure there are no moves in the move buffer. The transmission protocol is the same as for uploading. During downloading, the letter "d" will appear at the left of the teach pendant display and the number of moves will be constantly updated at the right of the display. After transmission is complete, the system will do an automatic "Hard Home" followed by a "Soft Home".

10. THE STALL ROUTINE

When your robot cannot carry out your command, cannot run the full number of steps you have given it with the move keys, the robot goes into a STALL and puts that word into the teach pendant display. Most commonly, this occurs in PLAY, LEARN, or EDIT when the robot runs into an obstruction which stops further motion. Somestimes, the robot runs into its base or the table top or even into itself.

Fortunately, you don't have to worry. The system is self correcting. After the display shows STALL, you will notice that the robot reverses direction and backs off from the obstacle it had hit. All you need to do is press the END key. The display will show the motor on which the stall had occurred and the corrected step count. For example, if the "D" motor had stalled at 180 steps, then your count after the STALL might show D 150, because you had lost 30 steps by backing up.

Occasionally, a STALL occurs during RUN, especially if you had not tested your program with the EDIT routine by stepping rhough the program one step at a time. You might have misjudged the actual path of the robot in executing the move sequence and run into an unexpected obstruction.

If this occurs, the display will show the move in which the STALL condition occurred. Press END to escape the stall trap. The display will ask you to do a "Hard Home". Follow the instruction and the display will change to "GO" and will find its own soft home and go into PLAY mode.

The advantage of the STALL routine is that the system remembers the program it had learned. For instance, if a stall occurs during RUN mode, the system continues with the move sequence after you have escaped the stall by pressing END and then doing a "Hard Home". However, if the motors cannot move as directed during EDIT mode or while the XR is doing a GO HOME or "Hard Home", then you can only escape by pressing the RESET, which essentially loses the program.

II. KEY FUNCTIONS

A. **SET HOME.** The SET HOME key stores a SOFT HOME or software home position for the robot. The SOFT HOME position can be the same as HARD HOME or can be established at some point closer to the final position of the robot. It is the position from which the user begins a move sequence.

The SOFT HOME position will remain until the power is turned off or the reset button is pressed. Before using the SOFT HOME key, put the system into PLAY mode and do a HARD HOME (see SHIFT/HOME below).

B. **GO HOME.** This key sends the robot to the SOFT HOME position. The GO HOME command, like the DELAY command does not have to be followed by the ENTER key. The GO HOME command is usually used at the end of a move sequence. The user must make sure that the moves made will not cause arm interference when the GO HOME instruction is executed.

C. **LEARN.** This key sets the teach pendant into LEARN mode. In LEARN mode, an "L" will be in the leftmost display of the pendant. In this mode the robot can be taught a move sequence. Unless the memory has been cleared (see SHIFT/END), new moves will be added to the last move in the prior sequence. The added instructions may not accurately reflect the position of the arm, so be careful. To begin LEARN, set the system in PLAY mode, do a HARD HOME and set a SOFT HOME.

D. **END**. This key is the last step to complete the LEARN, EDIT, RUN and STALL modes. When in LEARN mode, the END key puts the system into PLAY mode. If by chance your sequence contains 148 moves, the teach pendant display will indicate that memory is now FULL and you will have no choice except to press the END key which will take you out of LEARN mode.

In EDIT mode, the END key returns the system to PLAY mode. In STALL mode the END key takes you out of the stall. In RUN mode the END key stops the robot when the current move cycle has been completed.

E. **ENTER.** This key is active only in LEARN mode to record a robot position. For each position the user can move up to 8 motors. The user can make corrections as needed before pressing the ENTER key. The move as entered will be the algebraic sum of all the plus moves and minus moves for each motor.

Every time that a move is ENTERed, the display of the last move is cleared out and the total number of moves currently in the buffer is displayed on the right side of the display.

The maximum number of move entries allowed is 149 including all DELAYs and GO HOMEs.

F. **RUN**. This key can be activated only in PLAY mode after a move sequence has been LEARNed and the robot has returned to the HOME position. When this key is pushed, the letter"r" will appear in the leftmost position of the display. While the robot is in RUN mode the rightmost displays will specify the number of the current move. If the entry is a DELAY, the display will so indicate. If the RUN key is pressed a second time, the robot will HOLD and continue running after the RUN key is pressed a third time. To stop the RUN, press the END key.

G. **STORE.** The STORE key is used to upload a move sequence to the host computer for storage on disk. The system must be in PLAY mode in order to be able to begin the upload. Put the disk with the UPLOAD/DOWNLOAD program in the disk drive and make the appropriate responses as they appear on the menus. See APPENDIX 1.

NOTE: During uploading a letter "U" will appear in the leftmost display.

H. **EDIT**. The EDIT mode permits altering LEARNed motor moves, one at a time. To enter the EDIT mode, the robot must be at the HOME position, and the system must be in PLAY mode.

When the EDIT key is pressed, the robot makes its first move in the sequence, and the letter "E" appears in the leftmost display. When the move is complete, the rest of the display contains the count for H motor in the first move. For example, if the the count for H motor is zero for the first move, the display should read: "EH 0000".
In the first example, "E" indicates the EDIT mode, "H" indicates that H motor is being addressed, and the five remaining displays are the count for H motor. Using the FWD key, it is possible to step through motors from "H" to "A", or vice versa using the BACK key. The FWD and BACK key do not go beyond the current move (see FWD and BACK.)

EDITing a move for a certain motor is possible only while that motor is being addressed on the display. A DELAY move can either be inserted or deleted during EDIT.

I. **FWD**. This key moves the EDIT function from H motor to G, F, E, D, C, B, and A one at a time, each time it is pressed. The sequence runs from accessory motors to the base of the robot and up to the gripper. This allows the proper positioning if the larger components before the hand and gripper are adjusted. After the A motor is addressed, the FWD key becomes inactive. At that time you can move back over the motors from "A" through "H" or press the STEP key to get to the next move.

J. **BACK**. This key is identical to FWD except that the motor moves are displayed from "A" to "H".

K. **STEP**. While in EDIT mode, this key will bring you to the next move in the sequence. It is best to STEP through the entire sequence before going into RUN.

L. **DELAY**. This key is used in LEARN mode to enter a DELAY in the move sequence. When the sequence is run, the DELAY will generate a pause of approximately 1.5 seconds.

Multiple DELAYs can be entered by pressing the key as many times as needed. RUNning a move sequence with multiple delays can make the system appear "hung up". The appearance of the word "dELAy" in the display should tell the user that system is not really hung-up.

M. **1**. The '1' key will translate each press of the 14 move keys (see MOVE KEYS) into one encoder position move of the motor or roughly .9 degrees. This is useful for fine robot control. Although the robot moves slowly during positioning, it will move with normal speed when the move sequence is RUN.

N. **10**. The '10' key which is the default key on power-up or reset, is identical to the '1' key except that it causes a motor to move 10 encoder positions for each key press. The "10" key allows normal, easy movement of the robot.

O. **100**. The '100' key moves motors 100 encoder positions and is convenient for making long moves during PLAY, LEARN, or EDIT while using the carousel or the conveyor.

P. **MOVE KEYS.** The 16 keys on the right half of the TEACH PENDANT which operate the robot are the motor move keys. When the MOVE keys are pressed, the length of each move may be modified by depressing the '1', the '10', and the '100' keys. Holding any key down invokes the AUTO REPEAT feature.

The "A" motor (gripper) keys operate differently from those of the others . There are only two positions for the grip: "CLOSEd" or "OPEn". The "CLOSEd" key is inactive if the grip is already closed. Similarly the "OPEn" key is inactive if the grip is already open. The "CLOSEd" state is achieved when the motor "A" is stalled and stopped. To grasp an object with the gripper, simply press the key on the right to close and when you are ready to release, press the other gripper key. The system takes care of the rest.

When the other 14 motor move keys are pressed, the display shows the distance moved by that motor. The system constantly checks for stall conditions. If a stall occurs, the system automatically backs up the motor and corrects it. Then the word "StALL" appears on the display, and the user must press END to get out of the STALL trap.

CAUTION: User should **AVOID** using the MOVE KEYS corresponding to any motors which are not connected. Failure to observe proper precaution can hang up the system, causing possible transistor damage, and, of course, causes you to lose your program.

12. SHIFT KEY FUNCTIONS

These are the two finger functions that combine the SHIFT key, bottom left with another function key from the two left columns of the teach pendant. You should get used to keeping your left thumb somewhere near the SHIFT key. The second key is pressed while the SHIFT key is held down.

A. SHIFT/ GO HOME. This combination performs the HARD HOME function which orients the robot to its environment. When you first turn on the power or do a reset, the robot goes to the position defined by the cams on its three major axes, the shoulder cam, the elbow cam, and the waist. The system maintains the hand position as before and sets the gripper to OPEN.

Because the hand pivot does not alter, it is suggested that the user carefully position the robot hand. Many people start with the arm in a horizontal position and the hand pivot in a vertical position before doing the HARD HOME. After the HARD HOME position has been the system will simply send the robot to that position without searching for the cams.

B. SHIFT/ END. This key combination clears the memory. It is usually used when you first enter the LEARN mode to make sure that a sequence will not contain any unplanned moves. This key is active only in PLAY mode. It should be used with care.

C. SHIFT/STORE. This key is used to download information through the center serial port. The format of data and communication set up would be same as in STORE (upload). During the course of the download a letter "d" appears in the leftmost display. As the moves are received by the system, the count display will be updated on the right side of the display. Instructions are provided on the UPLOAD/DOWNLOAD diskette, see Appendix I.

D. **SHIFT/ FWD.** This key is active only in EDIT mode to INSERT a move or a DELAY. The move or DELAY is always inserted before the current move in EDIT buffer and the process is as follows: When SHIFT/FWD is pressed, the current move in the buffer will be RUN back and the word "InSrt" appears on the display. This means that a blank move has been inserted provided that the buffer is not yet full. If the user presses the DELAY key the blank move will become a DELAY. This is the only time in EDIT mode that DELAY key is active. As soon as any other key is pressed, the DELAY key becomes inactive.

E. **SHIFT/ BACK.** This key is used to DELETE moves or DELAYS, and it is active only during the EDIT mode. The process is as follows: Upon pressing SHIFT/BACK, the current move in the EDIT buffer is RUN back, and the word "dELEt" appears in the right side of the display momentarily. Then the next move in the sequence will be RUN and and the prior move is deleted.

The advanced user may want to experiment with compressing moves if he notices that the buffer is close to full and the move sequence is not complete. The user should single step through the program in EDIT to see whether any moves could be combined. After selected adjacent moves are collapsed then the SHIFT/BACK can be used to delete the extra moves.

F. **SHIFT/ STEP.** This key is the mirror image of STEP. Whereas STEP moves the robot forward through a sequence, the SHIFT/STEP moves the robot backwards. It is active only during EDIT mode.

13. PENDANT DISPLAYS

The little round, red indicator in the upper right hand corner of the pendant keyboard indicates that power is available at the pendant.

The following table shows the list of all messages that will appear on the display when some event takes place:

DISPLAY	EVENT	LOCATION
dO hArd	User is forced to do a HARD HOME.	Whole display
nO dAtA	RUN or EDIT or STORE is pressed while there is no sequence in buffer.	Whole display
dO GO	User is requested to go to Soft Home	5 rightmost
SEt	SOFT HOME is set.	3 rightmost
GO	Go to SOFT HOME.	3 rightmost
hArd	Seeking HARD HOME.	4 rightmost
StALL	One of the motors is stalled and reversed in PLAY, or LEARN, or EDIT.	5 rightmost
Hold	RUN key was pressed twice	4 rightmost
FULL	148 or more moves in buffer. Press GO HOME or END.	4 rightmost
dELAy	A delay is entered in LEARN, or, addressed in RUN or EDIT.	5 rightmost
dELEt	A move was deleted	5 rightmost

DISPLAY	EVENT	LOCATION
InSrt	A bank move was inserted	5 rightmost
LAST	Currently editing last move which may not be edited.	4 rightmost
0-127	The total number of moves, or the number of currently addressed move	1-3 rightmost
1	'1' key pressed, steps set to 1.	rightmost
10	'10' key pressed, steps set to 10.	2 rightmost
100	'100' key pressed, steps set to 100.	3 rightmost
t-P 2-0	The current version of the teach pendant. (teach pendant 2.0)	Whole display
CLOSEd	Grip closed.	6 rightmost
OPEn	Grip opened.	4 rightmost.
P	Currently in PLAY mode.	Leftmost
L	Currently in LEARN mode.	Leftmost
E	Currently in EDIT mode.	Leftmost
r	Currently in RUN mode.	Leftmost
U	Currently UPLOADING	Leftmost
d	Currently DOWNLOADING	Leftmost

NOTE: Occasionally some combinations of key presses will give your displays other than those listed. For further information about key press combinations, call R&D department of Rhino Robots at 217-352-8485.

For the time being, when you run into some other display which is not explained in this manual, press RESET and start over. If you have a combination teach pendant/controller unit, you will have only one RESET switch to press. If you have separate controller units, you will have to press RESET on each unit.

14. WARRANTY AGREEMENT

MECHANICAL AND ELECTRONIC

RHINO ROBOTS, INC. hereinafter refered to as a SELLER, makes the following warranties, which are exclusive.

SELLER guarantees the equipment, known as the RHINO XR SERIES TEACH PENDANT against imperfections and manufacturing defects for a period of one (1) year from the date of purchase for all mechanical parts and one hundred and eighty (180) days for all electronic parts, provided that said equipment is properly installed and operated according to SELLER's specifications and instructions.

SELLER's sole obligation under the above paragraph shall be limited to replacing or repairing parts which prove defective at the time of installation, or within the specified warranty periods stated.

SELLER hereby disclaims liability for any loss, injury, or harm which occurs as a result of the use of said equipment; including but not limited to personal injury or business and commercial loss.

SELLER disclaims any warranty of fitness for any particular purpose whatsoever with respect to the goods being sold, and buter acknowledges that he is not relying on the SELLER's skill or judgement to select or furnish goods suitable for any particular purpose.

The SELLER has made no warranty that the goods covered by this contract are merchantable.

SELLER has made no affirmation of fact and has made no promise relating to the goods being sold which has become part of the basis of the bargain made or has created or amounted to an express warranty that the goods would conform to any such affirmation or promise.

SELLER makes no representations as to whether the goods being sold are free of the rightful claim of any third party by way of infringement from patent, trademark, or other property rights in the goods and disclaims any warranty against infringement with respect to such goods.

There are no warranties which extend beyond the description on the face of this contract.

NOTE: In order for the warranty to be valid, the warranty card must be compeleted and returned within **30** days.

For all service information, call Rhino Robots at 217-352-8485.

APPENDIX I.

INSTRUCTIONS FOR "UPLOAD" AND "DOWNLOAD"

A diskette is provided for UPLOAD/DOWNLOAD with the teach pendant. A separate disk is available with software for each of the 3 following personal computers:

1. APPLE II
2. IBM PC
3. Radio Shack TRS-80

A description of the cable required for UPLOADing or DOWNLOADing, and instructions to operate the program for each computer follow:

1. <u>APPLE II.</u> The cable required for using the APPLE II program is the same kind of RS-232C cable which is used to interface the teach pendant and the XR series controller. The pin-outs are as follows:

```
                 TEACH PENDANT        APPLE II
Male DB 25                                            Male DB 25
Transmit           2 <-------------------> 2          Receive
Receive            3 <-------------------> 3          Transmit
Ground             7 <-------------------> 7          Ground
CTS                4 <-|               |-> 4          RTS
                       |               |
RTS                5 <-|               |-> 5          CTS

DTR                8 <-|               |-> 8          DCD
                       |               |
NC                 20<-|               |->20          DTR
```

INSTRUCTIONS:

1. Turn off your computer and install your serial card in slot #2 of the Apple.
2. Connect the cable between the center port of the teach pendant and the serial card connector.
3. Insert the provided disk in the drive 1, and turn on the power. (Boot up the system).
4. Specify the type of the card that you installed in slot #2. (The program requires that 1 of the 3 specified cards be used.)
5. Specify the action (Upload/Download) that will take place.
6. Follow the program instructions to Upload or Download.

2. **IBM PC.** The required cable for using the IBM PC program has the following pin-outs:

```
                  TEACH PENDANT         IBM PC
Male DB 25                                              Female DB 25
Transmit              2 <-------------------> 3           Receive
Receive               3 <--------------------> 2          Transmit
Ground                7 <--------------------> 7          Ground
CTS                   4 <-|               |-> 4           RTS
                          |               |
RTS                   5 <-|               |-> 5           CTS*

DTR                   8 <-|               |-> 8           DCD
                          |               |
NC                   20<-|                |->20           DTR
```

INSTRUCTIONS:

1. Connect the IBM cable between the center port of the teach pendant and the serial port of the IBM PC (on the back panel).
2. Insert the provided diskette in drive A. Turn on the power and system will be booted up.
3. Enter the current date and time.
4. Program will be automatically loaded to the memory and run.
5. Specify the action (Upload/Download) that will take place.
6. Follow the program instructions to Upload or Download.

Note: The program automatically appends a ".MOV" extension to all the filenames which are entered by the user. Therefore the user should only enter the first part of the filename.

3. <u>TRS-80.</u> The required cable for using the TRS-80 program has the following pin-outs:

```
                TEACH PENDANT          TRS-80
Male DB 25                                              Male DB 25
Transmit                  2 <-------------------> 3     Receive
Receive                   3 <-------------------> 2     Transmit
Ground                    7 <-------------------> 7     Ground
CTS                       4 <-|             |-> 4       RTS
                              |             |
RTS                       5 <-|             |-> 5       CTS
                                            |
DTR                       8 <-|             |-> 8       DCD
                              |
NC                       20<-|
DSR                       6 <-------------------> 20    DTR
```

INSTRUCTIONS:

1. Connect the TRS-80 cable between the center port of the teach pendant and the serial port of the TRS-80 (at the bottom). The connector which goes to the computer should be a 90 degree connector, or the computer should have a supporter under it so that it can be held up for the regular connector.
2. Before placing the diskette in the drive, turn on the power and wait for both drive lights to go off.
3. Insert the diskette provided in drive 0 (lower drive) and press the reset key. The system will be booted up.
4. Enter the current date and time.
5. Program will be automatically loaded to the memory and run.
6. Specify the action (Upload/Download) that will take place.
7. Follow the program instructions to Upload or Download.

APPENDIX II.

SOFTWARE LISTINGS FOR UPLOAD/DOWNLOAD

The following software listings for Upload/download programs are provided on the diskette.

The programs for APPLE II and TRS-80 utilize some machine code routines in addition to the BASIC programming. The IBM PC program is completely written in ADVANCED BASIC .

```
10  REM ************************
20  REM *   RHINO ROBOTS, INC.  *
30  REM *   P.O. BOX 4010       *
40  REM *   CHAMPAIGN, IL 61820 *
50  REM *   (217) 352-8485      *
60  REM *    UPLOAD/DOWNLOAD    *
70  REM *     APRIL 1, 1984     *
80  REM *    BY  FOAD REKABI    *
90  REM ************************
100  REM BASIC LANGUAGE PROGRAM
110  REM TO ASSIST USER WITH UP-
120  REM LOADING AND DOWNLOADING
130  REM USING AN APPLE COMPUTER
140  REM WITH CHOICE OF 3 INTER-
150  REM FACE CARDS.
160  REM
165  ONERR  GOTO 2000
170  GOSUB 3000
180  VTAB 7: HTAB 7: PRINT : HTAB 7: PRINT "YOU HAVE A CHOICE OF USING"
185 : HTAB 7: PRINT "THREE DIFFERENT INTERFACE"
190  HTAB 7: PRINT "CARDS WITH THIS PROGRAM AND"
195  HTAB 7: PRINT "YOUR APPLE COMPUTER:"
200  PRINT : PRINT : HTAB 7: PRINT "1...RHINO-COM LANGUAGE CARD"
210  HTAB 7: PRINT "2...CCS 7710A CARD"
220  HTAB 7: PRINT "3...SUPER SERIAL CARD"
230  VTAB 20: HTAB 11: PRINT "YOUR CARD? (1-3) ";: GET C$: VTAB 20: HTAB 28: PRINT C$
240  IF C$ = "1" THEN N$ = "UPDOWNLC.0": GOTO 300
250  IF C$ = "2" THEN N$ = "UPDOWN77.0": GOTO 350
260  IF C$ = "3" THEN N$ = "UPDOWNSS.0": GOTO 400
270  GOTO 230
300 H$ = "  LANGUAGE CARD  ": GOTO 500
350 H$ = " CCS 7710A  CARD ": GOTO 500
400 H$ = "SUPER SERIAL CARD": GOTO 500
500  GOSUB 3000
505  VTAB 12: HTAB 4: PRINT "LOADING ";H$;" DRIVER"
510  PRINT  CHR$ (4) + "BLOAD" + N$ + ",A$6000"
520  HOME : GOSUB 3000: VTAB 8: HTAB 3: PRINT "DO YOU WANT TO UPLOAD OR DOWNLOAD"
530  HTAB 3: PRINT "A MOVE SEQUENCE?": PRINT : PRINT : HTAB 8
540  PRINT "1...UPLOAD FROM PENDANT": HTAB 8: PRINT "2...DOWNLOAD TO PENDANT"
550  VTAB 16: HTAB 11: INPUT "YOUR CHOICE? (1/2) ";C$
```

```
560  IF C$ = "1" THEN  GOTO 670
570  IF C$ = "2" THEN  GOTO 590
580  GOTO 550
590  HOME : PRINT  CHR$ (4) + "CATALOG": PRINT : HTAB 4
600  INPUT "WHICH FILE FOR DOWNLOADING? ";F$: PRINT  CHR$ (4) + "BLOAD" + F$ + ",A$8000"
605  GOSUB 3000
610  VTAB 5: HTAB 7: PRINT "TO DOWNLOAD A MOVE SEQUENCE": PRINT : HTAB 4
620  PRINT "HOLD DOWN 'SHIFT', PRESS 'STORE'": HTAB 9: PRINT "ON HANDHELD KEYBOARD,"
630  HTAB 5: PRINT "THEN TYPE '2' ON HOST KEYBOARD"
640  CALL 24576: POKE  - 16368,0
650  PRINT : PRINT : HTAB 10: PRINT "SEQUENCE DOWNLOADED"
655  PRINT : HTAB 3: PRINT "CHECK BY PRESSING 'RUN' ON PENDANT"
660  GOTO 730
670  GOSUB 3000: VTAB 6: HTAB 7: PRINT "TO UPLOAD A MOVE SEQUENCE"
680  PRINT : HTAB 5: PRINT "TYPE '1' ON HOST KEYBOARD THEN"
690  HTAB 8: PRINT "PRESS 'STORE' ON PENDANT"
700  CALL 24576: POKE  - 16368,0
710  PRINT : PRINT : HTAB 11: PRINT "SEQUENCE UPLOADED": PRINT : HTAB 10
720  INPUT "NAME FOR SAVING FILE ";F$: PRINT  CHR$ (4) + "BSAVE" + F$ + ",A$8000,L$C00"
730  PRINT : HTAB (09): PRINT "TYPE 'RUN' TO RESTART"
740  INPUT "]";R$: IF R$ = "RUN" THEN  GOSUB 3000: GOTO 520
750  END
2000  HOME : VTAB 12: HTAB 16: PRINT "TRY AGAIN": FOR I = 1 TO 1000: NEXT I: GOTO 520
3000  PRINT : HOME : HTAB 14: PRINT "RHINO ROBOTS"
3010  HTAB 8: PRINT "UPLOAD/DOWNLOAD DISKETTE": HTAB 14: PRINT "VERSION 2.0": RETURN
```

```
            TTL "UP/DOWNLOADER XR-2"
*
*
*       UPLOAD/DOWNLOAD
*       JORG  R JEMELKA
*       MARCH  08  1983
*          COPYRIGHT
*        RHINO  ROBOTS
*
*
*          MODIFIED BY
*          FOAD REKABI
*       DECEMBER 16, 1983
*
* LATEST MODIFICATION: APRIL 1, 1984
*
*
           ORG $6000
           OBJ $800
KBD        EQU $C000            ;APPLE KEYBOARD
*
* IF USING CCS 7710A CARD THE ADDRESSES FOR
* THE UART ARE AS FOLLOWS:
*STAT  EQU $C0A0
*DATA  EQU $C0A1
*
*
* IN CASE OF USING RHINO-COM CARD,
* THE ADDRESSES ARE AS FOLLOWING:
*STAT      EQU $C23C            ;STATUS REGISTER
*DATA      EQU STAT+1           ;RHINO-COM DATA REG.
*COMM      EQU DATA+1
*CONT      EQU COMM+1
* THE FOLLOWING WHICH ARE ALSO PART OF THE
* PROGRAM ARE THE ADDRESSES FOR
* SUPER SERIAL CARD.
*
*
STAT       EQU $C0A9            ;SUPER SERIAL STATUS
DATA       EQU STAT+1           ;DATA REGISTER
COMM       EQU DATA+1           ;COMMAND REGISTER
CONT       EQU COMM+1           ;CONTROL REGISTER
STO        EQU $C010            ;APPLE KBD STROBE
*
* ZERO PAGE INDIRECT BASE REGISTERS (INDEX),Y
* ONE EACH FOR MOTOR SIGNS, LO MOVE AND HI MOVE BYTES
*
SBUFF      EPZ $FA
LBUFF      EPZ $FC
HBUFF      EPZ $FE
MOVES      EQU $88FF
CMASK      EQU $07
MAXMO      EQU $80
CLC        EQU 97
CR         EQU $8D
```

```
*
*  CV AND CH = APPLE CURSOR CONTROL LOCATIONS
*
CH       EPZ $24
CV       EPZ $25
*
*
*
*         START OF PROGRAM
*
*
* SET UP THE UART FOR SUPER SERIAL
* OR RHINO-COM CARD.
*
ALPHA    STA STAT
         LDA #$1E            ;8DATA, 1 STOP BIT, NO PARITY
         STA CONT
         LDA #$D9
         STA COMM
**********************
*
* IN ORDER TO SET UP THE UART FOR
* CCS 7710A REPLACE THE ABOVE 5
* LINES WITH THE FOLLOWING:
*
*ALPHA   LDA #3
*        STA STAT
*        LDA #$15
*        STA STAT
*
**********************
         JSR SETZP
         LDA #$00
         STA CNT             ;CLEAR COUNT
         JSR SGE
         JSR CARR
         LDA CNT             ;USER INPUT SAVED IN CNT
         CMP #$02
         BEQ LOOPD           ;IF 2 THEN DOWNLOAD TO TEACH PENDANT
         CMP #$01            ;IF NOT UPLOAD,
         BNE ALPHA           ;THEN START OVER
         JSR INCH            ;GET # OF MOVES
         STA MOVES           ;IN TEACH PENDANT
         STA CNT
         JSR $FDDA
         JSR CARR
LOOP     LDY #$00
LOOP1    JSR INCH
         STA (SBUFF),Y       ;STORE SIGN
         ORA #$80
         JSR $FDED
         JSR INCH            ;GET HI MOTOR
         STA (HBUFF),Y       ;AND STORE
         JSR $FDDA
         JSR INCH            ;GET LO MOTOR
         STA (LBUFF),Y       ;AND STORE
         JSR $FDDA
         INY
         CPY #$08            ;DO FOR 8 MO
```

```
          BCC LOOP1
          JSR ADDZP        ;SET UP FOR NEXT MOVE
          DEC CNT          ;DO FOR ALL MOVES
          BNE LOOP
*
*
FIN       RTS              ;RETURN TO DOS
*
*      DOWNLOAD TO TEACH PENDANT
*
LOOPD     JSR SETZP
          LDA MOVES
          STA CNT
          JSR OUTCH
          JSR $FDDA        ;ECHO ON SCREEN
          JSR CARR         ;START NEW LINE
          LDY #$00
          LDX #$00
DLY0      INY              ;DELAY FOR TEACH PENDANT
          BNE DLY0
          INX
          BNE DLY0
LOOPD1    LDA (SBUFF),Y    ;GET SIGN
          JSR OUTCH        ;SEND TO XR-2
          ORA #$80         ;ECHO TO SCREEN
          JSR $FDED
          LDA (HBUFF),Y    ;GET HI BYT
          JSR OUTCH        ;TELL XR-2
          JSR $FDDA        ;ECHO IT
          LDA (LBUFF),Y    ;GET LO BYT
          JSR OUTCH        ;TELL XR-2
          JSR $FDDA        ;ECHO IT
          INY
          CLC
          CPY #$08         ;DO FOR 8 MOTORS
          BCC LOOPD1
          JSR ADDZP
          LDY #$00
DLY1      INY              ;DELAY FOR TEACH
          BNE DLY1         ;PENDANT
          INX
          BNE DLY1
          DEC CNT          ;DO FOR ALL MOVES
          BNE LOOPD1
          RTS
*
*     SET INDIRECT INDEX REGISTERS TO BASE ADDRESSES
*     FOR SIGN, HI MOTOR BYTE, AND LOW  MOTOR   BYTE
*
SETZP     LDA SBL
          STA SBUFF
          LDA SBH
          STA SBUFF+1
```

```
         LDA LBL
         STA LBUFF
         LDA LBH
         STA LBUFF+1
         LDA HBL
         STA HBUFF
         LDA HBH
         STA HBUFF+1
         RTS
*
*   UPDATE Z-PAGE INDIRECT INDEX BASE LOCATIONS (ADD 8)
*
ADDZP    CLD
         LDA SBUFF
         CLC
         ADC #$08
         STA SBUFF
         LDA SBUFF+1
         ADC #$00
         STA SBUFF+1
ADDA     LDA LBUFF
         CLC
         ADC #$08
         STA LBUFF
         LDA LBUFF+1
         ADC #$00
         STA LBUFF+1
ADDB     LDA HBUFF
         CLC
         ADC #$08
         STA HBUFF
         LDA HBUFF+1
         ADC #$00
         STA HBUFF+1
ADDC     RTS
*
* SUPER SERIAL OR RHINO-COM CARD TRANSMIT ROUTINE
*
OUTCH    PHA
OUT1     LDA STAT
         AND #$10
*  FOR CCS 7710A REPLACE THE ABOVE
* LINE WITH:       AND #$02
         BEQ OUT1
         PLA
         STA DATA
         RTS
*
* SUPER SERIAL OR RHINO-COM CARD RECEIVE ROUTINE
*
INCH     LDA STAT
         AND #$08
*  FOR CCS 7710A REPLACE THE ABOVE
* LINE WITH:       AND #$01
```

```
         BEQ INCH
         LDA DATA
         RTS
*
* SGE CLEARS SCREEN PUTS PROMPT ON SCREEN AND
* ROUTES THE PROGRAM ACCORDING TO KEYBOARD IN-
* PUT FROM USER VALID COMMANDS ARE "1" OR "2"
*
SGE      LDA #$08
         STA CV
         JSR CARR
         LDA #$08
         STA CH
         LDA #$00
         STA STO
SGE2     LDA KBD
         BPL SGE2
         JSR $FDED
         AND #$7F
         CLC
         CMP #$31
         BCC SGE
         CLC
         CMP #$33
         BCS SGE
         SEC
         SBC #$30
         STA CNT
         RTS
*
*  SKIP LINE ON SCREEN
*
CARR     LDA #CR
         JSR $FDED
         RTS
*
CNT      DFS 1
MESSO    STR "UPLOAD OR DOWNLOAD? (1/2) "
*
*  LO AND HY BASE ADDRESSES FOR SIGN, LO MOTOR BYTE
*  AND HI MOTOR BYTES OF RHINO MOVE SEQUENCE
*
SBL      BYT $8000
SBH      HBY $8000
LBL      BYT $8400
LBH      HBY $8400
HBL      BYT $8800
HBH      HBY $8800
*
         END
```

```
10 ' ***************************************************************
20 ' *                                                             *
30 ' *                    RHINO ROBOTS, INC.                       *
40 ' *                    P.O. BOX 4010                            *
50 ' *                    CHAMPAIGN, IL  61820                     *
60 ' *                    (217) 352-8485                           *
70 ' *                                                             *
80 ' *                    UPLOAD/DOWNLOAD                          *
90 ' *                          BY                                 *
100 '*                      FOAD REKABI                            *
110 '*                                                             *
120 '***************************************************************
130 REM
140 REM   OPEN COMMUNICATION LINK
150 REM
160 KEY OFF
170 CLOSE : OPEN "COM1:9600,N,8,1,CS,DS,CD" AS #1
180 ON COM(1) GOSUB 310:COM(1) OFF
190 DIM MV$(128):ON ERROR GOTO 1080
200 GOSUB 1000
210 LOCATE 7,3:PRINT "NUMBER","CHOICE"
220 LOCATE 8,3:PRINT "------","------"
230 LOCATE 10,3:PRINT "  1   ","UPLOAD FROM TEACH PENDANT"
240 LOCATE 11,3:PRINT "  2   ","DOWNLOAD TO TEACH PENDANT"
250 LOCATE 12,3:PRINT "  3   ","END THE PROGRAM"
260 LOCATE 14,3:PRINT "PUSH A NUMBER (1-3)"
270 A$=INKEY$:IF A$="" THEN LOCATE 6,70:PRINT TIME$:GOTO 270
280 IF A$<>"1" AND A$<>"2" AND A$<>"3" THEN 270
290 ON VAL(A$) GOSUB 310,570,300:GOTO 200
300 CLS:KEY ON:END
310 REM
320 REM UPLOAD
330 REM
340 GOSUB 1000
350 LOCATE 7,38:PRINT "UPLOAD"
360 LOCATE 9,3:PRINT "PUSH 'X' NOW TO GO BACK TO MAIN MENU, OR"
370 LOCATE 10,3:PRINT "PUSH 'U' FIRST, FOLLOWED BY 'STORE' ON THE PENDANT."
380 A$=INKEY$:IF A$="" THEN LOCATE 6,70:PRINT TIME$:GOTO 380
390 IF A$="X" THEN RETURN
400 IF A$<>"U" THEN 380
410 BUF$="":J=0:FL=0:COM(1) ON:LOCATE 20,33:PRINT "PLEASE STAND BY...":LOCATE 22
,28:PRINT "CURRENTLY RECEIVING MOVE # "
420 IF FL=0 THEN LOCATE 6,70:PRINT TIME$:LOCATE 22,53:PRINT J-1:GOTO 420
430 I=9:GOSUB 560:I=10:GOSUB 560:I=20:GOSUB 560:I=22:GOSUB 560
440 L$=LEFT$(MV$(0),1)
450 LOCATE 9,4:PRINT "BESIDES THE HOME BUFFER ";ASC(L$)-1;" MOVES WERE RECORDED.
"

460 LOCATE 12,3:PRINT "PLEASE ENTER THE FILE NAME TO SAVE THE MOVE SEQUENCE, OR"

470 LOCATE 13,3:PRINT "ONLY HIT THE <RETURN> KEY TO IGNORE THE MOVE SEQUENCE."
480 LOCATE 15,1:PRINT "CURRENT MOVE FILES: ";:FILES "*.MOV"
490 LOCATE 23,5:INPUT "FILE NAME (.MOV): ";A$:IF A$="" THEN RETURN
```

BEST COPY AVAILABLE

```
500 A$=A$+".MOV":I=ASC(L$)
510 OPEN A$ AS #2 LEN=24
520 FIELD #2,24 AS SD$
530 LSET SD$=L$:PUT #2,1
540 FOR I=1 TO J-1:LSET SD$=MV$(I):PUT #2:NEXT
550 CLOSE 2:RETURN
560 LOCATE I,1:FOR I=1 TO 8:PRINT "                ";:NEXT:RETURN
570 REM
580 REM DOWNLOAD
590 REM
600 GOSUB 1000
610 LOCATE 7,37:PRINT "DOWNLOAD"
620 LOCATE 8,1:PRINT "MOVE FILES:"
630 LOCATE 9,1:FILES "*.MOV"
640 LOCATE 20,1:PRINT "ENTER THE FILE NAME TO DOWNLOAD, OR PRESS <RETURN> KEY TO
 EXIT."
650 LOCATE 22,5:INPUT "FILE NAME (.MOV): ";A$
660 IF A$="" THEN RETURN ELSE A$=A$+".MOV"
670 I=20:GOSUB 560:I=22:GOSUB 560
680 LOCATE 18,3:PRINT "FILE: ";A$
690 OPEN A$ AS #2 LEN=24
700 FIELD #2,24 AS SD$
710 GET #2,1:L$=LEFT$(SD$,1):J=ASC(L$)
720 IF J>0 THEN 770
730 CLOSE 2:KILL A$:LOCATE 20,5:PRINT A$;" DOES NOT EXIST."
740 LOCATE 22,5:PRINT "HIT ANY KEY TO CONTINUE."
750 B$=INKEY$:IF B$="" THEN LOCATE 6,70:PRINT TIME$:GOTO 750
760 I=18:GOSUB 560:LOCATE 22,20:PRINT "                    ":GOTO 640
770 LOCATE 20,3:PRINT "ON THE PENDANT WHILE PUSHING THE 'SHIFT' KEY, PUSH THE 'S
TORE' KEY,"
780 LOCATE 21,1:PRINT "THEN PUSH 'D' ON THE KEYBOARD"
790 B$=INKEY$:IF B$<>"D" THEN LOCATE 6,70:PRINT TIME$:GOTO 790
800 I=20:GOSUB 560:I=21:GOSUB 560
810 PRINT #1,L$;
820 LOCATE 20,1:PRINT "BESIDES THE HOME BUFFER,THERE ARE ";J-1;" MOVES IN ";A$
830 LOCATE 23,33:PRINT "PLEASE STAND BY...":LOCATE 24,31:PRINT "TRANSMITTING MOV
E #";
840 FOR I=1 TO J:GET #2:LOCATE 6,70:PRINT TIME$:LOCATE 24,50:PRINT I;
850 FOR FL=1 TO 10:LOCATE 6,70:PRINT TIME$:NEXT
860 PRINT #1,SD$;:NEXT:CLOSE 2
870 LOCATE 23,30:PRINT "TRANSMISSION COMPLETE      "
880 LOCATE 24,25:PRINT "HIT ANY KEY TO GO BACK TO MENU.";
890 B$=INKEY$:IF B$<> "" THEN RETURN
900 LOCATE 6,70:PRINT TIME$:GOTO 890
910 REM
920 REM  COLLECT DATA FROM TEACH PENDANT
930 REM
940 IF FL=0 THEN BUF$=""
950 WHILE NOT EOF(1):W$=INPUT$(LOC(1),#1):BUF$=BUF$+W$:WEND
960 IF BUF$="" THEN RETURN
970 MV$(J)=BUF$:J=J+1
    IF J=ASC(LEFT$(MV$(0),1))+1 THEN COM(1) OFF:FL=1
    RETURN
```

BEST COPY AVAILABLE

```
1000 REM
1010 REM  SCREEN HEADING
1020 REM
1030 CLS : LOCATE 2,33 :PRINT "RHINO ROBOTS INC."
1040 LOCATE 3,27 :PRINT "TEACH PENDANT UPLOAD/DOWNLOAD"
1050 LOCATE 4,31 :PRINT "IBM PERSONAL COMPUTER"
1060 FOR I=1 TO 40:LOCATE 5,41-I:PRINT "=";:LOCATE 5,40+I:PRINT "=";:NEXT I:PRIN
T
1070 RETURN
1080 REM
1090 REM ERROR HANDLING
1100 REM
1110 PRINT CHR$(7);
1120 IF ERR=71 THEN LOCATE 24,1:PRINT "DISK NOT READY, PLACE THE DISKETTE IN DRI
VE AND HIT ANY KEY TO CONTINUE.";:GOTO 1180
1130 IF ERR=70 THEN LOCATE 24,1:PRINT "DISK WRITE PROTECTED, CHANGE YOUR DISKETT
E AND HIT ANY KEY TO CONTINUE.";:GOTO 1180
1140 IF (ERR=53) AND (ERL=480) THEN LOCATE 15,1:PRINT "NO MOVE FILES ON DISK.":R
ESUME 490
1150 IF ERR=61 THEN LOCATE 24,1:PRINT "DISK FULL, CHANGE YOUR DISKETTE AND HIT A
NY KEY TO CONTINUE.";:GOTO 1180
1160 IF ERR=53 THEN LOCATE 24,1:PRINT "NO MOVE FILES ON DISK, CHANGE YOUR DISK A
ND HIT ANY KEY TO CONTINUE.";:GOTO 1180
1170 PRINT "ERROR #";ERR;" AT LINE #";ERL:END
1180 B$=INKEY$:IF B$="" THEN 1180
1190 LOCATE 24,1:FOR I=1 TO 79:PRINT " ";:NEXT:LOCATE 9,1
1200 RESUME
```

BEST COPY AVAILABLE

BEST COPY AVAILABLE

```
10 ''''''''''''''''''''''''''''''''''''''''''''''''''''
20 '
30 '            Rhino Robots, Inc.
40 '             3402 N. Mattis, Box 4010
50 '             Champaign, IL.  61820
60 '            Author: Thomas L. Redman.
70 '
75 ''''''''''''''''''''''''''''''''''''''''''''''''''''
80 DEFINT A-Z
90 DIM MVS(7,128)
100 CLS:DEFUSR0=&HC000:DEFUSR1=&H0040
110 GOSUB 1000:P$="":GOSUB 1060
120 PRINT:PRINT @725,;:INPUT "SELECT 1, 2, OR 3";P$
130 ON VAL(P$) GOSUB 300,400
150 IF VAL(P$) <> 3 THEN  110
160 END
300 GOSUB 1000:PRINT@594,;:INPUT "You wish to download (Y/N)";I$
302 IF LEFT$(I$,1)="N" THEN RETURN
304 IF LEFT$(I$,1)<>"Y" THEN GOTO 300
307 I$=""
310 PRINT@594,CHR$(30);:INPUT " Name the file is saved under";I$
315 ON ERROR GOTO 3000
320  OPEN"I",1,I$
330  INPUT#1,MVS(0,0)
340 FOR I=1 TO MVS(0,0)
350   FOR J=0 TO 7
360     INPUT#1,MVS(J,I)
380   NEXT J,I
383 CLOSE:ON ERROR GOTO 0
385 PRINT @584,CHR$(31);"Hold <sSHIFT> and press <STORE> on the Teach
387 PRINT @654,"When the 'd' appears at the teach pendant"
389 PRINT @717,"display, press the enter key at the TRS-80.":X=USR1(0:
390 PRINT @580,CHR$(31):PRINT @540,;
 IF MVS(0,0)<>0 THEN X=USR0(VARPTR(MVS(0,0)))
393 RETURN
400 GOSUB 1000:PRINT @596,;:INPUT "Do you wish to upload(Y/N)";I$
402 IF LEFT$(I$,1)="N" THEN RETURN
404 IF LEFT$(I$,1)<>"Y" THEN 400
406 PRINT @399,CHR$(31):PRINT@399,"Press <STORE> at the Teach Pendant.

410 MVS(0,0)=0:X=USR0(VARPTR(MVS(0,0)))
415 I$=""
420 GOSUB 1000:PRINT @592,;:INPUT "Name for saving file under";I$
422 IF I$="X" THEN RETURN
423 ON ERROR GOTO 2000
424 OPEN"O",1,I$
426 PRINT#1,MVS(0,0)
```

```
430 FOR I=1 TO MVS(0,0)
440  FOR J=0 TO 7
450    PRINT#1,MVS(J,I)
460  NEXT J
470 NEXT I
475 CLOSE:ON ERROR GOTO 0
480 RETURN
1000 'CLEAR SCREEN AND DISPLAY MAIN MENU.
1010 CLS
1020 PRINT@24,"RHINO ROBOTS,INC."
1030 PRINT@85,"DEMONSTRATION DISKETTE"
1040 PRINT@154,"VERSION 1.00"
1050 PRINT:PRINT:RETURN
1060 PRINT TAB(15),"1-DOWNLOAD TO TEACH PENDANT."
1070 PRINT:PRINT TAB(15),"2-UPLOAD TO TRS-80 FROM TEACH PENDANT."
1075 PRINT:PRINT TAB(15),"3-END PROGRAM."
1080 RETURN
2000 GOSUB 1000
2010 PRINT @456,"No disk space is available. Make sure the drive"
2020 PRINT @520,"door is closed and there is space on the diskette."
2030 I$="":PRINT @655,;:INPUT "Continue (press the enter key)";I$
2040 RESUME 400
3000 GOSUB 1000
3010 PRINT @456,"The file was not found. Please make sure there is"
3020 PRINT@520,"a diskette in the drive and that it contains the"
3030 PRINT@584,"requested file."
3040 I$="":PRINT @719,;:INPUT "Continue (press the enter key)";I$
3050 RESUME 300
3060 END
```

```
00100 MAX       DEFL      128
00110 VID       DEFL      4041H    ;LOCATION TO POKE THE MOVE NUMB.
00120 BASRET    DEFL      0A9AH
00130           ORG       0C000H-128
00140           DEFS      128
00150 NWSTK     CALL      0A7FH
00160           LD        (MVARR),HL         ;HL<-LOCATION OF ARRAY.
00170           DI
00180           LD        (SPSAVE),SP        ;SAVE BASIC STACK POINTER.
00190           LD        SP,NWSTK
00200           LD        A,0FDH   ;DISABLE COMMUNICATIONS.
00210           OUT       (0EAH),A
```

BEST COPY AVAILABLE

BEST COPY AVAILABLE

```
00220          LD      A,'M'
00230          CALL    033H
00240          LD      A,'O'
00250          CALL    033H
00260          LD      A,'V'
00270          CALL    033H
00280          LD      A,'E'
00290          CALL    033H
00300          LD      A,' '
00310          CALL    033H
00320          LD      A,(HL)
00330          LD      C,A
00340          INC     C
00350          CP      00
00360          JR      Z,UP      ;JUMP TO UPLOAD ROUTINE.
00370 ;
00380 ;DOWN LOAD TO THE TEACH PENDANT.
00390 ;
00400 DOWN     CALL    OUTCH
00410          LD      B,A
00420          LD      DE,10H
00430          ADD     HL,DE
00440 ;OUTER LOOP. LOOPS 1NCE FOR EACH SET OF MOVES
00450 LOOPD    PUSH    BC
00460          CALL    UPDIS
00470          LD      B,08
00480 ;INNER LOOP. ONCE FOR EACH MOTOR FOR EACH SET OF MOVES.
00490 LOOP     LD      E,(HL)
00500          INC     HL
00510          LD      D,(HL)
00520          BIT     7,(HL)
00530          INC     HL
00540          JR      Z,NCMP  ;NEGATIVE MOVE IF NZ..
00550          LD      A,D
00560          CPL
00570          LD      D,A
00580          LD      A,E
00590          CPL
00600          LD      E,A
00610          INC     DE
00620          LD      A,'-'
00630          CALL    OUTCH
00640          LD      A,B
00650          CP      01
00660          JR      NZ,SENDV
00670 ;A MOTOR ONLY HAS MOVES OF LENGTH ZERO.
00680 ;OPEN OR CLOSED STATE DETERMINED BY SIGN.
00690          XOR     A
00700          LD      D,A
00710          LD      E,A
00720          JP      SENDV
```

```
00730 NCMP     LD       A,'+'    ;POSITIVE MOVE.
00740          CALL     OUTCH
00750 ;SEND MOVE.
00760 SENDV    LD       A,D
00770          CALL     OUTCH
00780          LD       A,E
00790          CALL     OUTCH
00800          DJNZ     LOOP
00810          POP      BC
00820          DJNZ     LOOPD
00830          JP       RTRN
00840 ;
00850 ;UPLOAD FROM TEACH PENDANT TO HOST.
00860 ;
00870 UP       CALL     INCH
00880          LD       (HL),A
00890          LD       C,A
00900          INC      C
00910          LD       B,A
00920          CP       00
00930          JR       Z,RTRN
00940          LD       DE,10H   ;INCR PTR TO NEXT ROW OF MATRIX.
00950          ADD      HL,DE
00960 ;
00970 ;OUTER LOOP. LOOP ONCE FOR EACH SET OF MOVES.
00980 ;
00990 LOOPU1   PUSH     BC
01000          CALL     UPDIS
01010          LD       B,08H
01020 ;INNER LOOP. LOOP ONCE FOR EACH MOVE OF EACH SET OF MOVES
01030 LOOPU    CALL     INCH
01040          LD       (AREG),A
01050          CALL     INCH
01060          LD       D,A
01070          CALL     INCH
01080          LD       E,A
01090          LD       A,(AREG)
01100          CP       2BH
01110          JR       Z,NCOMP ;NEGATIVE IF NZ.
01120 GOON     LD       A,E
01130          CPL
01140          LD       E,A
01150          LD       A,D
01160          CPL
01170          LD       D,A
01180          INC      DE
01190 ;A MOTOR IS + OR - ZERO.
01200          LD       A,B
01210          CP       01
01220          JR       NZ,NCOMF
01230          LD       A,0FFH
01240          LD       D,A
01250          LD       E,A
```

BEST COPY AVAILABLE

BEST COPY AVAILABLE

```
01260 NCOMP   LD      (HL),E
01270         INC     HL
01280         LD      (HL),D
01290         INC     HL
01300         DJNZ    LOOPU
01310         POP     BC
01320         DJNZ    LOOPU1
01330 ;THIS CODE IMPLEMENTS A RETURN TO BASIC.
01340 RTRN    LD      HL,(SPSAVE)
01350         LD      SP,HL
01360         JP      BASRET
01370 SPSAVE  DEFW    0
01380 MVARR   DEFW    0
01390 AREG    DEFS    1         ;STORES THE ASCII SIGN CHAR.
01400 ;
01410 ;GET A CHARACTER FROM THE TEACH PENDANT.
01420 ;
01430 INCH    PUSH    DE
01440         LD      A,0FFH
01450         OUT     (0EAH),A
01460 INCH1   IN      A,(0EAH)
01470         BIT     7,A
01480         JR      Z,INCH1
01490         IN      A,(0EBH)
01500         PUSH    AF
01510 ;DISSABLE COMMUNICATIONS. HANDSHAKE LINE TUGED ON.
01520         LD      A,0FDH
01530         OUT     (0EAH),A
01540         POP     AF
01550         POP     DE
01560         RET
01570 ;
01580 ;SEND A CHARACTER TO THE TEACH PENDANT.
01590 ;
01600 OUTCH   PUSH    DE
01610         PUSH    BC
01620         PUSH    AF
01630 OUTLP   IN      A,(0EAH)
01640         BIT     6,A
01650         JR      Z,OUTLP
01660         POP     AF
01670         OUT     (0EBH),A
01680         CALL    DELAY   ;ALLOW DELAY.
01690         POP     BC
01700         POP     DE
01710         RET
01720 ;
01730 ;DELAY ROUTINE.
01740 ;
01750 DELAY   PUSH    BC
01760         LD      B,04H
01770 DL1     PUSH    BC
01780         LD      B,0FFH
01790 DL2     NOP
01800         DJNZ    DL2
```

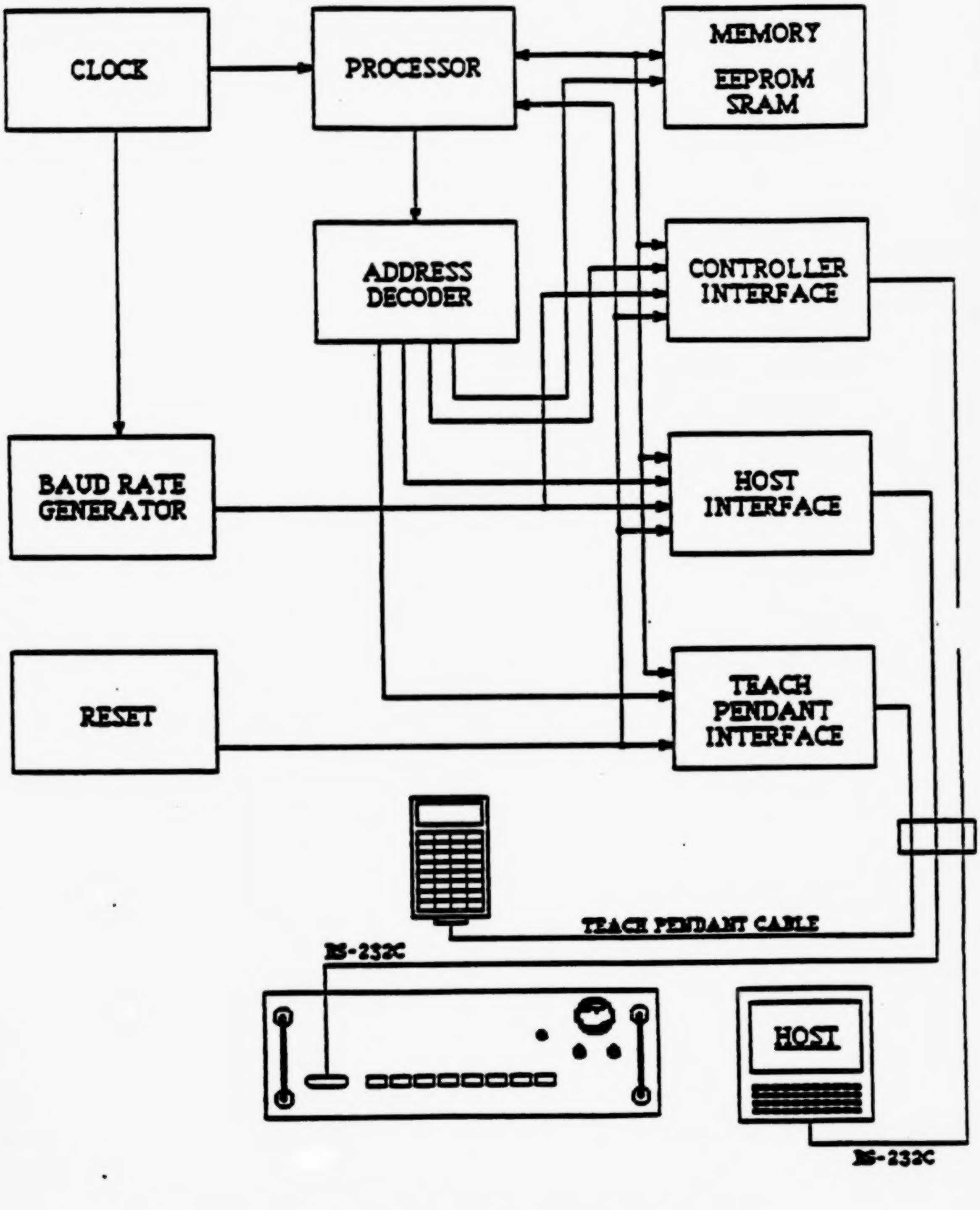
```
01810          POP     BC
01820          DJNZ    DL1
01830          POP     BC
01840          RET
01850 ;
01860 ;DISPLAY THE NUMBER OF MOVES ON THE SCREEN.
01870 ;
01880 UPDIS    PUSH    BC
01890          EXX
01900          XOR     A
01910          LD      (TEN),A
01920          LD      (ONE),A
01930          LD      (HUND),A
01940          LD      IY,HUND
01950          POP     BC
01960          LD      A,C
01970          SUB     B
01980 ;DETERMINE THE NUMBER FOR THE HUNDREDS COLUMN.
01990 HNDS     CP      064H
02000          JP      M,TNS
02010          INC     (IY)
02020          SUB     064H
02030          JR      HNDS
02040 TNS      LD      IY,TEN
02050 ;DETERMINE THE NUMBER FOR THE TENS COLUMN.
02060 TNSO     CP      0AH
02070          JP      M,ONS
02080          INC     (IY)
02090          SUB     0AH
02100          JR      TNSO
02110 ;DETERMINE THE NUMBER FOR THE ONES COLUMN.
02120 ONS      LD      IY,ONE
02130 ONSO     CP      00H
02140          JR      Z,DISPLY
02150          INC     (IY)
02160          DEC     A
02170          JR      ONSO
02180 ;NOW CONVERT DECIMAL TO ASCII AND DISPLAY
02190 DISPLY   LD      A,(HUND)
02200          AND     0FH
02210          ADD     A,30H
02220          CALL    033H
02230          LD      A,(TEN)
02240          AND     0FH
02250          ADD     A,30H
02260          CALL    033H
02270          LD      A,(ONE)
02280          AND     0FH
02290          ADD     A,30H
02300          CALL    033H
02310          LD      IY,4020H
02320          LD      (IY),21H
02330          EXX
02340          RET
02350 TEN      DEFS    1         ;STORES VALUE FOR TENS COLUMN.
02360 ONE      DEFS    1         ;STORES VALUE FOR ONES COLUMN.
02370 HUND     DEFS    1         ;STORES VALUE FOR HUNDREDS COLUMN.
02380          END
```

APPENDIX III.

DRAWINGS AND SCHEMATICS

The following are provided for your use:

1. A block diagram of the teach pendant system.
2. A schematic of the teach pendant computer board along with a description of the main components.
3. A schematic of the teach pendant pendant board and a description of the main elements.


CLOCK
PROCESSOR
MEMORY
EEPROM
SRAM
ADDRESS
DECODER
CONTROLLER
INTERFACE
BAUD RATE
GENERATOR
HOST
INTERFACE
RESET
TEACH
PENDANT
INTERFACE
TEACH PENDANT CABLE
RS-232C
HOST
RS-232C

BLOCK DIAGRAM OF THE TEACH PENDANT SYSTEM

TEACH PENDANT VERSION 2.0

2. THE TEACH PENDANT COMPUTER BOARD, schematic below. A description of the main components follows.

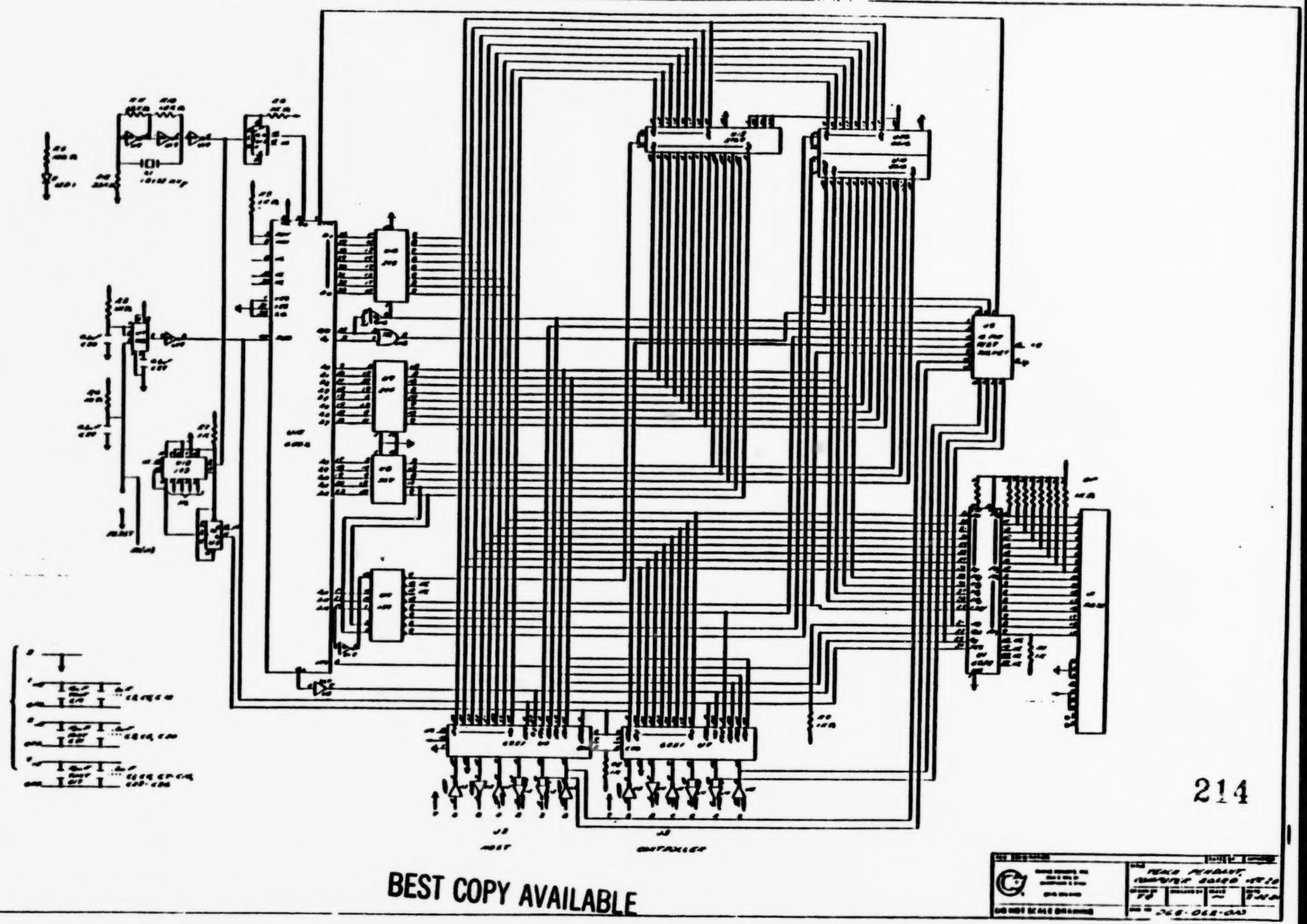
2.1 CLOCK CIRCUIT. The clock circuit provides a 1.8432 MHz reference timing signal to the BAUD rate generator and the same signal divided by 2 to the 6502 CPU. The circuit consists of a 1.8432 MHz crystal (X1), three inverters (U19) and a 74LS76 flip-flop (U12).

2.2 BAUD RATE GENERATOR. The BAUD rate generator takes the 1.8432 MHz signal from the clock generator, divides it by 6 with a 74LS193 (U18) and divides it again by 2 (squaring the signal) with a 74LS75 (U12) to get a 16 x 9600 BAUD signal. The signal is then routed to the external clock and receiver clock inputs of both UARTS.

2.3 CONTROLLER INTERFACE. The controller interface provides the serial communication link to the XR controller. It consists of a 6551 UART (U7) and line drivers (1488, U4 and 1489, U5) feeding the J3 DB25 connector. The UART operates at a fixed 9600 BAUD, transmitting on line 3 of J3. Lines 4 and 5 of J3 should be jumpered together at the connector as well as line 6 to 8 of J3.

2.4 HOST COMPUTER INTERFACE. The host computer interface provides the serial communication link to a host computer. It consists of a 6551 UART (U6) and line drivers (1488, U2 and 1489, U3 feeding the J2 DB25 connector. The UART operates at a fixed 9600 BAUD, transmitting on line 2 of J2 and receiving on line 3 of J2. Lines 4 and 5 of J2 should be jumpered together at the connector as well as line 6 to line 8 of J2.

2.5 PENDANT INTERFACE. The pendant interface provides the parallel communication link to the hand pendant. It consists of a 6522 VIA (U1) and a DB25 (J1) connector.


HOST
CONTROLLER
TEACH PENDANT,
COMPUTER BOARD
DO NOT SCALE DRAWING

BEST COPY AVAILABLE

2.6 RESET CIRCUIT. The reset circuit consists of a 555 (U17) timer, 1 inverter (U19) and two switches. One switch is located on the front panel of the case. The other is on the pendant itself. The reset circuit controls the processor, both UARTS, and the VIA.

2.7 MEMORY. Memory consists of 8K by 8 of UV Erasable Programmable Read Only Memory (2764 EPROM, U13) and 4K by 8 of Static Random Access Memory (4016 SRAM, U14 and U20).

2.8 ADDRESS DECODER. The address decoder consists of one inverter (U19) and a 74LS139 dual 2-to-4 line decoder (U11).

2.9 PROCESSOR UNIT. The processor unit consists of a 1MHz 6502 CPU (U15), a 74LS245 (U10) data bus transceiver, a 74LS244 (U9) and 74LS367 (U8) address buffer, and a 74LS32 (U16) for buffering the read/write line and to gate the read/write line with the processor 01 clock line for the static RAM.

3. TEACH PENDANT PENDANT BOARD, schematic below.
A brief description of the main elements follows.

3.1 KEYPAD The pendant's operation continuously shifts between two distinct modes, the key input mode and the display mode.

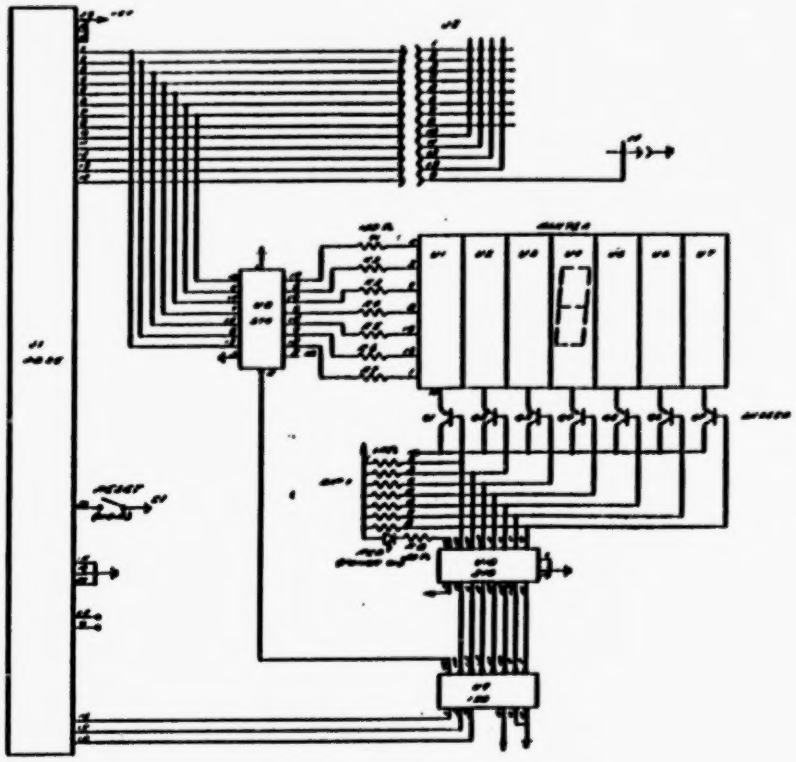
3.2 KEY INPUT MODE. In this mode, the computer is scanning the 8 by 4 matrix of membrane switches on the pendant. Thirteen lines from the computer's 6522 VIA are used to accomplish this function. Eight of the lines are inputs to the 6522 with one line per row of the matrix. Four lines are outputs from the 6522 with one line per column of the matrix. The 13th line is an input to the 6522 and is connected to the SHIFT key of the matrix.

During operation of the key input mode, no more than one of the 4 output lines is low. This low line continuously rotates among the 4 output lines while the 8 input lines are read by the 6522. When a key is pressed, one of the 8 input lines becomes connected to one of the 4 output lines. If that outpuut line is low, then oone of the 8 input lines is also low. By decoding the 8 input lines in relation to the 4 output lines the value of the key being pressed can be determined.

In other words, each key can be thought of as a point in an X,Y coordinate system with the 8 input lines as the Y coordinate and the 4 output lines as the X coordinate. Hence, each key has its own unique X,Y coordinate.

The SHIFT key is detected separately on its own input line.

Only the key switch matrix is used for this mode on the pendant.



DO NOT SCALE DRAWING

TEACH PENDANT, PENDANT BOARD [illegible]

TG

~

066-068-011

BEST COPY AVAILABLE

3.3 DISPLAY MODE. The display mode is controlled by 10 output lines from the 6522 VIA. Seven of the lines represent the data to be displayed and the other 3 lines control the latching of the data and also determine which of the seven 7-segment displays will be enabled.

This is accomplished by first presenting the data to be displayed to the pendant, then taking the 3 control lines to a low level, which enables the latching function. Then the 3 control lines are taken to the address required to enable the appropriate display, which completes the latching at the same time.

Note that, in order to avoid possible shorting together of the 6522's pins, the order in which the direction of the lines is changed must prevent it from occurring.

The computer considers the key input mode to be the normal mode. That is, the 9 row input lines are configured as inputs. When an interrupt occurs to refresh the display, the 4 output lines to the columns of the key matrix are first reconfigured as inputs before the 8 row input lines are configured as outputs. On completion of the display mode, the procedure is reversed.

A 74LS374 (U8) is used to latch the data with the latching input coming from a 74LS138 (U9). The 74LS138 also provides a 1 of 7 signal to a 74LS240 (U10) that drives a transistor, which, in turn, enables the associated 7-segment display (U1-U7).

APPENDIX IV.

TROUBLE SHOOTING

Loss of a signal can be due to a variety of reasons, including the following:

1. Open or shorted inputs.
2. Open or shorted outputs.
3. Open or shorted traces.
4. Missing parts.
5 Missing controlling signals.
6. Bad components.
7. Bent pins.

Most problems can be solved by visual inspection. Look for the solder bridges, unsoldered nodes, missing traces, missing parts, bent pins, shorted traces, and components that appear in less than good shape.

If a visual inspection does not show the source of the problem, try to determine where in the circuit the problem may be located and begin probing with an oscilloscope.

First probe the power input pins to be sure that power is getting to the board. It is usually a good idea to take a random sample of the power levels on several chips themselves. If no power is getting to the board, check all connections and the power supply. If only some chips are missing power, check for missing or bent pins or open traces.

If a signal is missing and the circuit board and its components look all right, a quick parts substitution may be all that is needed.

For additional information call the Rhino Robots hotline at 217-352-8485.

INSTRUCTIONS, MENUS, ETC., FOR
RHINO/APPLE SOFTWARE PROGRAM (RASP), VERSION 1.03

①

```
]RUN DEMO
          RHINO ROBOTS     DEMONSTRATION DISKETTE
          VERSION  0.3
           MAIN  MENU


        1...INSTRUCTIONS
        2...RUN PROGRAMS
        3...END  PROGRAM
       YOUR CHOICE? (1-3) 1
          RHINO ROBOTS     DEMONSTRATION DISKETTE
          VERSION  0.3
      INSTRUCTIONS AVAILABLE

       1...KEYBOARD   MOVE
       2...MOVE ALL JOINTS
       3...'TEACH' PROGRAM
       4...'XYZ'   PROGRAM
        TYPE (X) TO EXIT
       YOUR CHOICE? (1-4) X
          RHINO ROBOTS     DEMONSTRATION DISKETTE
          VERSION  0.3
           MAIN  MENU


        1...INSTRUCTIONS
        2...RUN PROGRAMS
        3...END  PROGRAM
       YOUR CHOICE? (1-3) 2
          RHINO ROBOTS     DEMONSTRATION DISKETTE
          VERSION  0.3
       PROGRAMS AVAILABLE

       1...KEYBOARD   MOVE
       2...MOVE ALL JOINTS
       3...'TEACH' PROGRAM
       4...'XYZ'   PROGRAM
        TYPE (X) TO EXIT
       YOUR CHOICE? (1-4) X
          RHINO ROBOTS     DEMONSTRATION DISKETTE
          VERSION  0.3
           MAIN  MENU


        1...INSTRUCTIONS
        2...RUN PROGRAMS
        3...END  PROGRAM
```

BEST COPY AVAILABLE

②

```
       YOUR CHOICE? (1-3) 1
           RHINO ROBOTS     DEMONSTRATION DISKETTE
           VERSION  0.3
     INSTRUCTIONS AVAILABLE

       1...KEYBOARD   MOVE
       2...MOVE ALL JOINTS
       3...'TEACH' PROGRAM
       4...'XYZ'   PROGRAM
        TYPE (X) TO EXIT
       YOUR CHOICE? (1-4) 1
           RHINO ROBOTS     DEMONSTRATION DISKETTE
           VERSION  0.3
       ONE MOMENT PLEASE
           READING DISK
           RHINO ROBOTS     DEMONSTRATION DISKETTE
           VERSION  0.3
   THIS PROGRAM IS WRITTEN TO
   DEMONSTRATE THE RHINO XR-1
   IN THE FIELD.  YOU CAN GET
   CONTINUOUS MOTION FROM ANY
   JOINT BY PRESSING THE  KEY
   THAT ACTIVATES IT AND  THE
   'REPT' KEY SIMULTANEOUSLY.
     PRESS ANY KEY TO CONTINUE
     TYPE B FOR BACK, X TO EXIT
                                RHINO ROBOTS     DEMONSTRATION DISKETTE

         VERSION  0.3
   THE KEYS THAT OPERATE THE IN-
   DIVIDUAL JOINTS ARE ARRANGED
   SO THAT THE JOINTS THAT HAVE
   A 'NATURAL' UP/DOWN MOVEMENT
   SUCH  AS THE SHOULDER  ELBOW
   OR HAND  HAVE  THEIR KEYS IN
   THE KEYBOARD'S TOPMOST  ROW.
     PRESS ANY KEY TO CONTINUE
     TYPE B FOR BACK, X TO EXIT
                                RHINO ROBOTS     DEMONSTRATION DISKETTE

         VERSION  0.3
   THE BASE IS OPERATED BY  THE
   'R' AND 'F'  KEYS.  PRESSING
   'R' ONCE MOVES  THE BASE  15
   STEPS TO YOUR LEFT (YOU  ARE
   FACING THE RHINO). SIMILARLY
   PRESSING 'F' MOVES THE RHINO
   15 STEPS TO YOUR RIGHT.
     PRESS ANY KEY TO CONTINUE
     TYPE B FOR BACK, X TO EXIT
                                RHINO ROBOTS     DEMONSTRATION DISKETTE

         VERSION  0.3
   REMEMBER THAT  PRESSING  THE
   'REPEAT' KEY AFTER  PRESSING
   ANY  LETTER  KEY THAT  MOVES
   A JOINT WILL GIVE CONTINUOUS
   MOVEMENT IN THAT JOINT.  THE
   SHOULDER JOINT IS  ACTIVATED
   USING THE 'T' AND 'G'  KEYS.
     PRESS ANY KEY TO CONTINUE
     TYPE B FOR BACK, X TO EXIT
                                RHINO ROBOTS     DEMONSTRATION DISKETTE
```

BEST COPY AVAILABLE

③

```
       VERSION  0.3
NOTE THAT THE 'T' KEY IS  IN
THE TOPMOST ROW OF KEYS. THE
'T'  KEY  MOVES THE SHOULDER
UP. THE 'G' KEY MOVES IT TO-
WARDS  THE TABLE TOP (DOWN).
NEXT  TO  THE SHOULDER  KEYS
ARE THE ELBOW KEYS. THEY ARE
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                      RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
THE 'Y' AND 'H' KEYS NEXT TO
THE SHOULDER KEYS.  PRESSING
THE  'Y' AND THE 'REPT' KEYS
GIVES CONTINUOUS UPWARD MOVE-
MENT OF THE ELBOW.  PRESSING
EITHER KEY ONCE GIVES YOU  A
15 STEP MOTION UP OR DOWN.
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                      RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
THE HAND ASSEMBLY IS OPERATED
USING THE  'U' AND  'J' KEYS.
'U' BEING IN THE  TOP  ROW OF
THE  KEYBOARD MOVES  THE HAND
ASSEMBLY UP. 'J' GIVES A DOWN
MOTION. THE FINGERS ARE ROTAT-
ED USING THE 'I' AND 'K' KEYS.
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                      RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
THE   'I'  KEY   ROTATES   THE
FINGERS IN A CLOCKWISE  MANNER
WHILE THE 'K'  KEY IS USED FOR
COUNTERCLOCKWISE ROTATION ....
AGAIN THE REPEAT KEY HELD DOWN
WITH  ANY JOINT  COMMAND  KEYS
GIVES YOU CONTINUOUS MOVEMENT.
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                      RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
THE FINGERS CAN BE OPENED
OR CLOSED  USING THE  'O'
OR 'L' KEYS.  THE 'L' KEY
CLOSES THE FINGERS  WHILE
THE  'O'  KEY OPENS THEM.
ALL MOTORS  HAVE SOFTWARE
PROTECTION AGAINST STALL.
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                      RHINO ROBOTS     DEMONSTRATION DISKETTE
```

```
        VERSION  0.3
THE G AND H  ACCESSORY MOTORS
HAVE MOTION THAT  IS  DEVICE-
DEPENDENT. EXPERIMENT TO DIS-
COVER THE MOVEMENT ASSOCIATED
WITH THE 'W'/'S'/'E'/'D' KEYS
AND THE ACCESSORIES YOU HAVE.
TO EXIT ALWAYS TYPE 'X'.
   PRESS ANY KEY TO CONTINUE
   TYPE B FOR BACK, X TO EXITRHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
   INSTRUCTIONS AVAILABLE

      1...KEYBOARD   MOVE
      2...MOVE ALL JOINTS
      3...'TEACH' PROGRAM
      4...'XYZ'   PROGRAM
       TYPE (X) TO EXIT
      YOUR CHOICE? (1-4) 2
         RHINO ROBOTS     DEMONSTRATION DISKETTE
         VERSION  0.3
      ONE MOMENT PLEASE
         READING DISK
         RHINO ROBOTS     DEMONSTRATION DISKETTE
         VERSION  0.3
THIS PROGRAM ALLOWS YOU TO
CONTROL ALL THE  MOTORS OF
THE RHINO BY GIVING EACH A
SPECIFIC NUMBER   OF STEPS
TO MOVE.  THE   FORMAT  FOR
THE COMMANDS IS  SIMPLE TO
USE BECAUSE YOUR INPUT  IS
   PRESS ANY KEY TO CONTINUE
   TYPE B FOR BACK, X TO EXIT
                                         RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
PROCESSED BY THE PROGRAM
TO MAKE IT USABLE BY THE
RHINO.  FOR  EXAMPLE  IF
YOU PRESS <RETURN>  ONLY
AT  A  MOTOR PROMPT THIS
WILL BE  INTERPRETED  AS
A +0000 ENTRY. SIMILARLY
   PRESS ANY KEY TO CONTINUE
   TYPE B FOR BACK, X TO EXIT
                                         RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
IF YOU ENTER -16 THEN THE
PROGRAM  STRUCTURES  YOUR
ENTRY  AS  IF IT HAD BEEN
-0016. EACH ENTRY MUST BE
FOLLOWED  BY  A <RETURN>.
TYPING THE CHARACTER  'X'
AS THE ONLY INPUT TERMIN-
   PRESS ANY KEY TO CONTINUE
   TYPE B FOR BACK, X TO EXIT
                                         RHINO ROBOTS     DEMONSTRATION DISKETTE
```

⑤

```
      VERSION  0.3
ATES THE PROGRAM. THE PRO-
GRAM ALLOWS YOU TO GET EX-
PERIENCE WITH THE  NUMBER
OF STEPS  ANY  PARTICULAR
JOINT REQUIRES TO MOVE IT
THROUGH SOME ANGLE. USING
THE PROGRAM  IS SUGGESTED
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                             RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
BEFORE YOU USE THE 'TEACH'
PROGRAM'S SEQUENCE EDITOR.
THE PROGRAM INITIALLY ASKS
FOR THE 'H' AND 'G'  MOTOR
STEPS. MOST LIKELY YOU MAY
NOT  HAVE  THE TRANSLATION
STAGE  OR  ROTARY TABLE TO
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                             RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
DEMONSTRATE. IN THIS CASE
SIMPLY TYPE A <RETURN> AT
THE 'H' AND 'G'  PROMPTS.
THE NEXT PROMTS ARE DESC-
RIPTIVE OF THE JOINTS RE-
QUIRING  INPUT  SUCH  AS
'BASE'  MOTOR  STEPS  OR
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                             RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
'SHOULDER' MOTOR STEPS ETC.
AT EACH OF THE PROMPTS JUST
ENTER THE  NUMBER  OF STEPS
YOU WANT THE  JOINT TO MOVE
REMEMBERING TO PRECEED EACH
NEGATIVE DIRECTION  COMMAND
WITH  A  '-'.  THE POSITIVE
  PRESS ANY KEY TO CONTINUE

  TYPE B FOR BACK, X TO EXIT
                             RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
DIRECTION INDICATOR '+' IS
NOT REQUIRED AS THE  MOTOR
DIRECTION DEFAULTS TO  '+'
IN THE ABSENCE OF AN EXPL-
ICIT 'MINUS' DIRECTION CO-
MMAND.  THE PROGRAM IS  OF
USE PRINCIPALLY AS A  DEM-
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                             RHINO ROBOTS     DEMONSTRATION DISKETTE
```

BEST COPY AVAILABLE

⑥

```
        VERSION  0.3
ONSTRATION  OF  INDIVIDUAL
JOINT  COMMANDS  THAT  ARE
THE EXECUTED TOGETHER.  IF
YOU  GIVE EACH JOINT  SOME
NUMBER OF STEPS..SAY '100'
AT THE PROMPTS.. AFTER YOU
HAVE  REPLIED  TO  ALL THE
   PRESS ANY KEY TO CONTINUE
   TYPE B FOR BACK, X TO EXIT
                                       RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
PROMTS THE PROGRAM WILL
EXERCISE ALL THE JOINTS
SIMULTANEOULSY.  IF YOU
INADVERTANTLY  MOVE THE
JOINTS INTO AN OBSTACLE
THE  PROGRAM  WILL  NOT
'HANG' SO DON'T WORRY.
   PRESS ANY KEY TO CONTINUE
   TYPE B FOR BACK, X TO EXITRHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
   INSTRUCTIONS AVAILABLE

     1...KEYBOARD   MOVE
     2...MOVE ALL JOINTS
     3...'TEACH' PROGRAM
     4...'XYZ'   PROGRAM
      TYPE (X) TO EXIT
     YOUR CHOICE? (1-4) 3
        RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
     ONE MOMENT PLEASE
        READING DISK
        RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
THE 'TEACH' PROGRAM CONSISTS OF
FOUR SECTIONS 'CREATE' 'MODIFY'
'SAVE' OR 'RUN' A MOVE SEQUENCE
MOVE SEQUENCES ARE CREATED WITH
THE SAME KEYPRESSES AS THE KEY-
BOARD  MOVE PROGRAM.  ONCE  YOU
HAVE POSITIONED  THE  RHINO  AT
   PRESS ANY KEY TO CONTINUE
   TYPE B FOR BACK, X TO EXIT
                                       RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
A SPOT THAT YOU WANT YOU CAN
ADD THAT STEP AS PART OF THE
MOVE SEQUENCE BY TYPING  'A'
THIS WILL  SAVE  THOSE MOTOR
STEPS AS PART OF A SEQUENCE.
YOU CAN HAVE UP TO 31 PARTS
IN A MOVE SEQUENCE. ONCE THE
   PRESS ANY KEY TO CONTINUE
   TYPE B FOR BACK, X TO EXIT
                                       RHINO ROBOTS     DEMONSTRATION DISKETTE
```

⑦

BEST COPY AVAILABLE

```
      VERSION  0.3
SEQUENCE IS CREATED YOU CAN
MODIFY IT RUN IT OR SAVE IT
THE MODIFY SAVE OR RUN SEC-
TIONS CAN BE USED WITH A SE-
QUNCE  THAT  IS  ALREADY ON
DISK OR A SEQUENCE YOU  ARE
IN THE PROCESS OF CREATING.
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                  RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
TO SAVE A SEQUENCE AFTER CRE-
ATING IT GET INTO THE 'SAVE'
PART OF THE PROGRAM. HERE YOU
HAVE THE CHOICE OF SAVING THE
SEQUENCE AS THE ONLY DISK SE-
QUENCE OR AS A PART OF A  SE-
QUNCE ON THE DISK.  IF  IT IS
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                  RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
TO BE ADDED TO A SEQUENCE ON
DISK IT WILL BE ADDED AT THE
END OF THE DISK SEQUENCE. IF
THERE ARE  NO  MOVES ON DISK
THEN  THE  CURRECNT SEQUENCE
WILL BE THE ONLY SEQUENCE IN
BOTH CASES.  TO EDIT A  MOVE
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                  RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
SEQUENCE GET INTO THE 'MODIFY'
SECTION.  HERE  YOU  CAN EDIT
EITHER THE SEQUENCE  YOU  HAVE
JUST CREATED  OR  THE SEQUENCE
ON DISK.  IN BOTH CASES THE SE-
QUNCE  WILL  BE DISPLAYED  ONE
STEP AT A TIME AND YOU WILL BE
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                  RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
PROMPTED WHETHER YOU WANT TO
MAKE CHANGES TO THE NUMBER OF
STEPS EACH MOTOR IS TO TRAVEL
IN THAT MOVE OF THE SEQUENCE.
IF YOU DECIDE TO MAKE CHANGES
THEN YOU WILL BE TAKEN TO ALL
THE MOTORS IN TURN AND BE IN-
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                  RHINO ROBOTS     DEMONSTRATION DISKETTE
```

```
        VERSION  0.3
VITED TO CHANGE EACH MOTOR'S
STEPS TO TRAVEL. ENTERING  A
SIMPLE <RETURN> AT ANY MOTOR
RETAINS THAT MOTOR'S PRESENT
VALUE.  CHANGES CAN BE  MADE
BY ENTERING  THE  NEW  STEPS
NEXT TO THE OLD AND PRESSING
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                           RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
<RETURN>.  TO SAVE THE MODIFIED
SEQUENCE ENTER THE SAVE SECTION
IF YOU ARE  EDITING  MOVES THAT
YOU ARE JUST CREATING OR IF YOU
ARE EDITING A DISK SEQUENCE THE
FINAL PROMPT WILL ASK IF  THESE
MOVES SHOULD BE SAVED  AT WHICH
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                           RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
TIME YOU SHOULD ANSWER 'Y' AT
THE PROMPT.  THE  RUN SECTION
ALLOWS YOU TO RUN EITHER  THE
MOVE SEQUENCE JUST CREATED OR
A SEQUENCE ON DISK.  YOU  CAN
REPEAT A SEQUENCE AS OFTEN AS
DESIRED WITH THE REPEAT FACTOR
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXITRHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
  INSTRUCTIONS AVAILABLE

    1...KEYBOARD   MOVE
    2...MOVE ALL JOINTS
    3...'TEACH' PROGRAM
    4...'XYZ'   PROGRAM
     TYPE (X) TO EXIT
    YOUR CHOICE? (1-4) 4
        RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
    ONE MOMENT PLEASE
        READING DISK
        RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
THE XYZ PROGRAM REQUIRES THAT
YOU SET THE LIMIT SWITCH CAMS
BY MOVING THE RHINO USING THE
KEYBOARD MOVE PROGRAM.  FIRST
THE CAM ON THE SHOULDER JOINT
SHOULD MAKE CONTACT WITH  THE
LIMIT SWITCH WHEN THE SHOULDER
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                           RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
IS 6.25 INCHES FORM THE REAR
OF THE BASE.  THE  ELBOW CAM
SHOULD TOUCH ITS SWITCH WHEN
THE ELBOW IS POSITIONED APPX
10 DEGREES BELOW HORIZONTAL.
THE HAND MUST POINT STRAIGHT
DOWN AND THE FINGERS SHOULD
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                  RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
BE CLOSED AND ALIGNED WITH
THE BASE. THE XYZ  PROGRAM
REQUIRES AS INPUT THE  CO-
ORDINATES IN MM OF THE POINT
THAT YOU WANT TO BE IN THE
FINGERTIP CENTRE.  FOR EX-
AMPLE A CO-ORDINATE INPUT
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                  RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
HAVING AN X OF 300 A Y OF
300 AND A Z OF 200  WILL
POSITION THE FINGERTIPS
300 MM TO THE RIGHT OF THE
RHINO'S CENTRE 300 MM AWAY
FROM THE RHINO'S CENTRE AND
200 MM UP FROM  THE TABLE
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                  RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
YOU CAN ALSO SPECIFY THAT
THE POINT BE REACHED WHEN
THE HAND IS NOT  POINTING
STRAIGHT DOWN BY GIVING THE
HAND ROTATE PROMPT SOME
ANGLE OTHER THAN ZERO (A
<RETURN> ALWAYS STANDS FOR
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                  RHINO ROBOTS     DEMONSTRATION DISKETTE
        VERSION  0.3
ZERO). IF YOU SPECIFY 90
DEGS THE HAND WILL BE A-
LIGNED PARALLEL  TO  THE
TABLE TOP.  SPECIFYING
135 DEGS  WILL  MAKE  IT
POINT 45 DEGS UP  AS  IF
GRASPING FROM BELOW. THE
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                                  RHINO ROBOTS     DEMONSTRATION DISKETTE
```

⑩

```
      VERSION  0.3
POINT REACHED BY THE FINGER-
TIPS WILL STILL BE  ACCURATE
WHEN MEASURED BUT THE APPRO-
ACH WILL BE FROM A DIFFERENT
ANGLE.  BE  CAREFUL  NOT  TO
DAMAGE THE  HAND'S  CIRCUIT
BOARD.  THE FINGERS WILL RE-
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                              RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
MAIN ALIGNED PARALLEL TO THE
BASE UNLESS YOU SPECIFY SOME
ANGULAR OFFSET.  THE FINGERS
CAN BE OPENED  OR  CLOSED AS
REQUIRED.  MOVES  SEQUENCES
THAT YOU CREATE CAN BE STORED
ON DISK AS  WITH  THE  JOINT
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                              RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
MOVE PROGRAM.  YOU CAN ALSO
EDIT A MOVE SEQUENCE.  THE
EDITOR EXECUTES THE MOVE AND
DISPLAYS THE VALUES OF THE
CO-ORDINATES. IF YOU WANT NO
CHANGE OF A MOVE IN EDIT MODE
YOU MUST SPACE OVER THE VALUE
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXIT
                              RHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
DISPLAYED ON THE SCREEN
USING THE RIGHT ARROW KEY
FOLLOWED BY <RETURN>.
CHANGES ART ENTERED BY OVER-
WRITING THE VALUES DISPLAYED
THE DIRECTION SIGNS (+/-)
ARE REQUIRED AS FIRST CHARACTER.
  PRESS ANY KEY TO CONTINUE
  TYPE B FOR BACK, X TO EXITRHINO ROBOTS     DEMONSTRATION DISKETTE
      VERSION  0.3
  INSTRUCTIONS AVAILABLE

    1...KEYBOARD   MOVE
    2...MOVE ALL JOINTS
    3...'TEACH' PROGRAM
    4...'XYZ'   PROGRAM
     TYPE (X) TO EXIT
    YOUR CHOICE? (1-4) X
       RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
        MAIN  MENU
```

BEST COPY AVAILABLE

⑪

```
     1...INSTRUCTIONS
     2...RUN PROGRAMS
     3...END  PROGRAM
   YOUR CHOICE? (1-3) 2
       RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
   PROGRAMS AVAILABLE

   1...KEYBOARD   MOVE
   2...MOVE ALL JOINTS
   3...'TEACH' PROGRAM
   4...'XYZ'   PROGRAM
    TYPE (X) TO EXIT
   YOUR CHOICE? (1-4) 1
       RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
   LOADING DRIVER
       COMMANDS ARE

   H  MOTOR KEYS = W,S
   G  MOTOR KEYS = E,D
   BASE     KEYS = R,F
   SHOULDER KEYS = T,G
   ELBOW    KEYS = Y,H
   HAND UP/DOWN  = U,J
   FINGERS ROT.  = I,K
   FINGERS O/C   = O,L
PRESS 'REPT' FOR LONG MOVES
     TYPE (X) TO EXIT
       RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
   PROGRAMS AVAILABLE

   1...KEYBOARD   MOVE
   2...MOVE ALL JOINTS
   3...'TEACH' PROGRAM
   4...'XYZ'   PROGRAM
    TYPE (X) TO EXIT
   YOUR CHOICE? (1-4) 2
       RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
      LOADING DRIVER
  PLEASE ENTER # OF STEPS

      H  MOTOR STEPS
      G  MOTOR STEPS
      BASE     STEPS
      SHOULDER STEPS
      ELBOW    STEPS
      HAND   UP/DOWN
      FINGER  ROTATE
      FINGER     O/C
FORMAT IS (+/-) 0..2000 STEPS
      TYPE (X) TO EXIT
```

⑫

```
       RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
    PROGRAMS AVAILABLE

    1...KEYBOARD   MOVE
    2...MOVE ALL JOINTS
    3...'TEACH' PROGRAM
    4...'XYZ'   PROGRAM
     TYPE (X) TO EXIT

    YOUR CHOICE? (1-4) 3
       RHINO ROBOTS     DEMONSTRATION DISKETTE
       VERSION  0.3
* — — — — — — — — — — — — —
         DO YOU WANT TO

  1..CREATE A MOVE SEQUENCE
  2..MODIFY A MOVE SEQUENCE
  3..SAVE   A MOVE SEQUENCE
  4..RUN    A MOVE SEQUENCE
        TYPE (X) TO EXIT
      YOUR CHOICE? (1-4) 1
         RHINO ROBOTS     DEMONSTRATION DISKETTE
         VERSION  0.3
      LOADING DRIVER
      H  MOTOR KEYS = W,S
      G  MOTOR KEYS = E,D
      BASE     KEYS = R,F
      SHOULDER KEYS = T,G
      ELBOW    KEYS = Y,H
      HAND UP/DOWN  = U,J
      FINGERS ROT.  = I,K
      FINGERS O/C   = O,L
     TYPE (A) TO ENTER MOVE

        MORE MOVES (Y/N) N
         RHINO ROBOTS     DEMONSTRATION DISKETTE
         VERSION  0.3

        DO YOU WANT TO

  1..CREATE A MOVE SEQUENCE
  2..MODIFY A MOVE SEQUENCE
  3..SAVE   A MOVE SEQUENCE
  4..RUN    A MOVE SEQUENCE
        TYPE (X) TO EXIT
      YOUR CHOICE? (1-4) 2
         RHINO ROBOTS     DEMONSTRATION DISKETTE
         VERSION  0.3
        LOADING DRIVER
         RHINO ROBOTS     DEMONSTRATION DISKETTE
         VERSION  0.3
1..MODIFY A SEQUENCE NOW ON DISK
2..MODIFY THE CURRENT SEQUENCE ONLY
      TYPE (X) TO EXIT
       YOUR CHOICE? (1-2) 1
   READ YOUR OR SYS FILE? (Y/S)
                                  YOUR FILENAME?
```

Note: From * above onward the "DO YOU WANT TO" parts will be repeated with listings or choices dependent upon what the programmer wishes to do.

BEST COPY AVAILABLE

```
]
]RUN HELLO
                RHINO ROBOTS
              RASP VERSION 3.0
              LOADING PROGRAM
                 RHINO ROBOTS
               RASP VERSION 3.0
                  MAIN  MENU

                 1 CLEAR MEMORY
                 2 LOAD  A FILE
                 3 SAVE  A FILE
                 4 HELP  SCREEN
                 5 RESUME  PROG
                 6 END  PROGRAM

             YOUR CHOICE? (1-6) )~1  RHINO ROBOTS
               RASP VERSION 3.0
                  MAIN  MENU

                 1 CLEAR MEMORY
                 2 LOAD  A FILE
                 3 SAVE  A FILE
                 4 HELP  SCREEN
                 5 RESUME  PROG
                 6 END  PROGRAM

             YOUR CHOICE? (1-6) )~2  RHINO ROBOTS
                RASP VERSION 3.0
                 MAIN PROGRAM

            FILENAME?

               LOADING


DISK VOLUME 254

*A 003 HELLO
*B 005 RESET.O
*B 017 CONTROL1.O
*B 002 FREAD.O
*A 003 READF
*B 063 CONTROL3
 B 014 MTEST
*B 002 LIMTEST
       PRESS ANY KEY TO CONTINUE
```

BEST COPY AVAILABLE

②

```
                 RHINO ROBOTS
              RASP VERSION 3.0
                  MAIN  MENU

                 1 CLEAR MEMORY
                 2 LOAD  A FILE
                 3 SAVE  A FILE
                 4 HELP  SCREEN
                 5 RESUME  PROG
                 6 END  PROGRAM

           YOUR CHOICE? (1-6) )~3  RHINO ROBOTS
              RASP VERSION 3.0
                 MAIN PROGRAM

        FILENAME?

              SAVING


DISK VOLUME 254

*A 003 HELLO
*B 005 RESET.O
*B 017 CONTROL1.O
*B 002 FREAD.O
*A 003 READF
*B 063 CONTROL3
 B 014 MTEST
*B 002 LIMTEST
        PRESS ANY KEY TO CONTINUE
                 RHINO ROBOTS
              RASP VERSION 3.0
                  MAIN  MENU

                 1 CLEAR MEMORY
                 2 LOAD  A FILE
                 3 SAVE  A FILE
                 4 HELP  SCREEN
                 5 RESUME  PROG
                 6 END  PROGRAM
```

```
      YOUR CHOICE? (1-6) )~4  RHINO ROBOTS
         RASP VERSION 3.0
            HELP  SCREEN

          MOTOR ABCDEFGH
          + KEY 23456789
          - KEY WERTYUIO

  MOTOR STEPS       HOME
  100  STEPS=L    SET   HOME=X
  10   STEPS=K    GO    HOME=S
  1    STEP =,    HARD HOME=J

   TEACH MODE       EDIT
  X=SET HOME    G=START EDIT
  C=LEARN MODE N=NEXT MOTOR
  V=ENTER MOVE H=LAST MOTOR
  D=END TEACH

        F= RUN     ^= DELAY

  PRESS ANY KEY TO CONTINUE
           RHINO ROBOTS
        RASP VERSION 3.0
            MAIN  MENU

           1 CLEAR MEMORY
           2 LOAD   A FILE
           3 SAVE   A FILE
           4 HELP   SCREEN
           5 RESUME   PROG
           6 END   PROGRAM

    YOUR CHOICE? (1-6) )~5  RHINO ROBOTS
       RASP VERSION 3.0
          MAIN PROGRAM

         MOTOR ABCDEFGH
         + KEY 23456789
         - KEY WERTYUIO

  V=ENTER            F=RUN
  ^=DELAY            G=EDIT
  S=GO HOME

  RETURN TO SUPERVISOR= B
        RHINO ROBOTS
      RASP VERSION 3.0
          MAIN  MENU

          1 CLEAR MEMORY
          2 LOAD   A FILE
          3 SAVE   A FILE
          4 HELP   SCREEN
          5 RESUME   PROG
          6 END   PROGRAM

    YOUR CHOICE? (1-6) )~6  RHINO ROBOTS
       RASP VERSION 3.0
         MAIN PROGRAM
```

BEST COPY AVAILABLE

Dr Ann

ROTERP: An Interpretive Language for Robot Control

High-level languages may help bridge the gap between artificial intelligence and the home experimenter's robot.

Gary Liming
1224 South Wheaton
St. Charles, MO 63301

A quick survey of the information being published on experimental robotics and artificial intelligence reveals a gap in practical information between the two subjects where there should be a bridge. Paradoxically, while most experiments in artificial intelligence are performed in high-level languages (notably LISP), most books and articles for robotics experimenters concentrate on hardware design and leave software control to assembly-language routines.

The assembly-language approach to robot control involves combining routines to produce a particular behavior in a robot. This method does have practical advantages: it requires little initial planning, it is usually memory-efficient, and the code can be entered by using simple switches or a keypad, making more expensive peripherals dedicated to the robot unnecessary. A major drawback, however, is the time and effort it takes to reorganize the routines to produce new behavior in the robot.

Perhaps the greatest disadvantage of using assembly language is the difficulty other experimenters have understanding the origins of a robot's behavior. Because assembly language is difficult to read, the underlying cause of the robot's behavior may elude even the well-versed experimenter. Duplicating the behavior of a robot with a different processor is usually even more difficult.

A microprocessor specifically designed for robotic control would resolve these problems. Such a microprocessor would use simple commands that exercise all of the robot's capabilities. A program for this microprocessor might look something like listing 1, which causes the robot to slowly walk a three-foot square. In addition to being easier to understand than assembly language, the program requires only 24 bytes of storage.

Enter the "Pseudoprocessor"

Our theoretical microprocessor would be practical only if all the components of the robot could be anticipated and mass production could justify its cost. In fact, custom processors have been used in intelligent instruments and even in sophisticated toys, but an experimental robot is a low-volume item and must be flexible enough to accommodate major new functions and devices. To expect all experimenters to use the same microprocessor is unrealistic.

Listing 1: *This example of a first attempt at a robotic language suggests how a robot-control language should look. Instructions are simple enough to be very flexible (they might be used in any number of situations) yet powerful enough so that the programmer doesn't get bogged down in details.*

```
0000  05          SLOW
0001  08          FEET
0002  30  0003    FORWARD 3
0005  34  0090    TURN RIGHT 90
0008  30  0003    FORWARD 3
0011  34  0090    TURN RIGHT 90
0014  30  0003    FORWARD 3
0017  34  0090    TURN RIGHT 90
0020  30  0003    FORWARD 3
0023  00          HALT
```

In lieu of a custom microprocessor, we might create a "pseudoprocessor" in software. Not only would it execute our robotic instructions, but it would be easy to extend or modify as well. This pseudoprocessor could be in the form of a small interpreter, possibly in PROM (programmable read-only memory). The interpreter

BEST COPY AVAILABLE

Single-Board Computer
—Microprocessor
—Limited available memory
—I/O interfaces

Output Devices
—Motors for forward, reverse, and turning motions
—Numeric display
—Bell or buzzer

Input Devices
—Proximity switches or ultrasonic sensors
—Numeric keypad
—Light detector

Table 1: *Attributes of a simple experimental robot.*

would fetch each instruction of the program from memory and call an appropriate routine to execute that instruction. The pseudoprocessor technique is not new; languages like Pascal and even some adventure-type games have been implemented for microprocessor systems using the concept.

If all of this sounds like an argument on behalf of the use of high-level languages, it is. A robotic pseudoprocessor has a number of advantages. First, it provides programming at a behavioral level that is easy for the programmer (or anyone else) to understand. And it creates programs that are less machine-dependent. It is also conducive to the creation of more powerful instructions than those on most microprocessors. Finally, it can be extended and modified, if designed properly.

A Simple Design

Let's see how the pseudoprocessor approach can be applied in a simple example. Table 1 shows the attributes of a basic experimental robot. It consists of an SBC (single-board computer) like many of those currently available, a frame and motors constructed with treads or wheels for mobility, some proximity sensors and light detectors, a bell or buzzer, and a numeric display and keypad (found on the SBC).

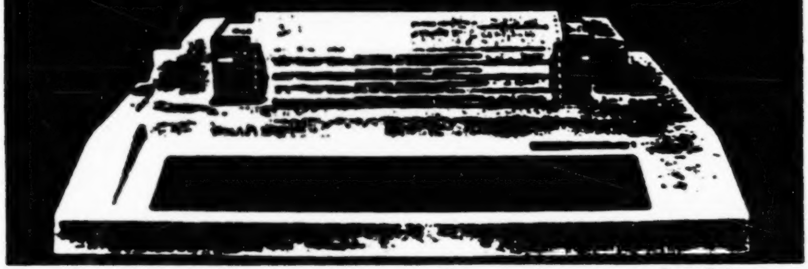
Given these simple peripheral devices, we can envision what a pseudoprocessor for a robot will look like. Of course, it will need instructions to turn the drive motors on and off and to acquire data from the sensors. It will have registers to count units of time and distance. Instructions that change the program flow and conditional testing will enhance its ability to make decisions. And it will include such standard items as a program counter, general-purpose registers, and I/O (input/output) instructions.

Figure 1 shows the architecture of the interpreter for the simple robot I named ROTERP. It has 26 general-purpose 16-bit registers, a program counter, condition codes, and some special-purpose registers. (I chose to have 26 general-purpose registers because they are easily represented by the letters of the alphabet.)

The array register, a location that points to the array being referenced, functions like an index register found on other processors. The speed register is a location for a number that the movement instructions use to determine a rate of speed. The distance-units register is a location that movement instructions use to set a scale for movement, and the time-units register specifies a time scale.

To produce a particular behavior, the registers are manipulated by the

Tune up your LA36

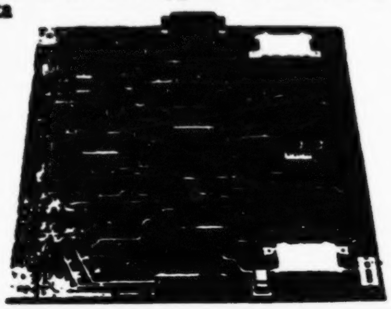


The DS120 Terminal Controller makes your LA36 perform like a DECwriter® III.

The Datasouth DS120 gives your DECwriter® II the high speed printing and versatile performance features of the DECwriter® III at only a fraction of the cost. The DS120 is a plug compatible replacement for your LA36 logic board which can be installed in minutes. Standard features include:

- 165 cps bidirectional printing
- Horizontal & Vertical Tabs
- Page Length Selection
- 110-4800 baud operation
- 1000 character print buffer
- X-on, X-off protocol
- Self Test
- RS232 interface
- 20 mA Current Loop interface
- Top of Form
- Adjustable Margins
- Double wide characters
- Parity selection
- Optional APL character set

Over 5,000 DS120 units are now being used by customers ranging from the Fortune 500 to personal computing enthusiasts. In numerous installations, entire networks of terminals have been upgraded to take advantage of today's higher speed data communications services. LSI microprocessor electronics and strict quality control ensure dependable performance for years to come. When service is required, we will respond promptly and effectively. Best of all, we can deliver immediately through our nationwide network of distributors. Just give us a call for all the details.



datasouth computer corporation
4216 Stuart Andrew Blvd. • Charlotte, North Carolina 28210 • 704/523-8500

Circle 141 on inquiry card.

instruction set given in table 2, which shows each instruction op (operation) code, its associated mnemonic, the type of operand it uses, and the number of bytes per instruction. Although the op codes are shown in hexadecimal, they could easily be given in decimal; hexadecimal was chosen only for convenience. The first 64 instructions control the robot's movement while the last 64 instructions control the processor registers. Because ROTERP uses 8-bit op codes, 128 possible instructions are still available for expansion.

In order to understand the origin of behavior in our simple robot, we need to look more closely at the instructions it follows. The first group of instructions, op codes 00 through 0F hexadecimal, are all 1-byte instructions. HALT simply returns control to the monitor ROM (read-only memory) on the SBC. ZERO initializes all the processor registers except the program counter. FAST, MEDIUM, and SLOW place a predetermined number in the speed register. Similarly, INCHES, FEET, and METERS store a value in the distance-units register, and MILLSEC, SECONDS, and HOURS set the time-units register. REPEAT places the starting address of the program into the program counter and normally causes it to loop indefinitely at the end of a program. RETURN signals the end of a subroutine.

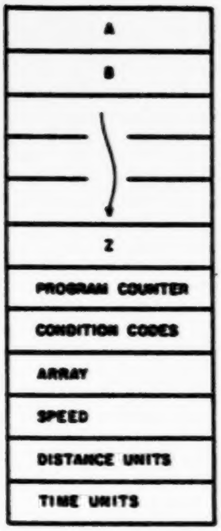


Figure 1: *Architecture of ROTERP, the interpreter for a simple robot. The interpreter handles 16-bit-wide registers (there is a general-purpose register for each letter of the alphabet as well as registers for condition codes and special purposes).*

THE
BENCHMARK®
A Word Processor
Worthy of Comparison . . .

FEATURE	BENCHMARK 2.0	Wordstar 3.21	BENCHMARK 3.0
Ease of Operation	YES	NO	YES
Plain English Commands	YES	NO	YES
Use Cursor Keys	YES	NO	YES
Control Characters	NO	YES	NO
Function Keys	YES	NO	YES
Computer Aided Tutorial	YES	YES	YES
Descriptive Directory	YES	NO	YES
29 Character Title	YES	NO	YES
Author & Operator ID	YES	NO	YES
Document Size	YES	NO	YES
Creation Date	YES	NO	YES
Revision Time & Date	YES	NO	YES
Standard Editing Features	YES	YES	YES
Go To Any Page	YES	NO	YES
Interactive Printing	YES	NO	YES
Automatic Reformatting	YES	NO	YES
Automatic Repagination	YES	NO	YES
Headings and Footings	YES	YES	YES
Multi-line	YES	NO	YES
Keyboard Phrases	YES	NO	YES
Proportional Printing	NO	NO	YES
Business Graphics	NO	NO	YES
Paragraph Assembly	NO	NO	YES
Edit Marking	NO	NO	YES
Auto Widow/Orphan Protect	NO	NO	YES
Auto Footnoting	NO	NO	YES
Built In Calculator	NO	NO	YES
Price	$249	$495	$499

DEMAND A COMPARISON!

Metasoft Corporation

Circle 286 on inquiry card.

The next group of instructions, 10 through 1F hexadecimal, are each 2 bytes long and use a single-byte operand to specify a general-purpose register that contains a working value. For instance, BEEP C, represented as 17 03, causes the robot to beep the number of times indicated by register C. FORWARD B (10 02) causes the robot to move forward the number of units defined in register B. Similarly, HESITATE A (19 01) causes the robot to wait A units of time, DISPLAY A displays register A, and ENTER A places a number from the keypad in register A. TEST tests the indicated register and sets the appropriate condition codes. RANDOM returns a random number and places it in the given register. SCAN and PROXIMITY read the respective

BEST COPY AVAILABLE

Op code	Mnemonic	Operand type	Bytes required	Op code	Mnemonic	Operand type	Bytes required	Op code	Mnemonic	Operand type	Bytes required	Op code	Mnemonic	Operand type	Bytes required
00	HALT		1	20	FORWARD	RC*	2	40	JUMP	R	2	60	SET	R,R	3
01	ZERO		1	21	REVERSE	RC	2	41	JUMPEQ	R	2	61	ADD	R,R	3
02				22	RIGHT	RC	2	42	JUMPNE	R	2	62	SUB	R,R	3
03	FAST		1	23	LEFT	RC	2	43				63	MUL	R,R	3
04	MEDIUM		1	24	RTURN	RC	2	44	SBROUTNE	R	2	64	DIV	R,R	3
05	SLOW		1	25	LTURN	RC	2	45	ARRAY	R	2	65			
06				26				46				66			
07	INCHES		1	27				47				67	COMPARE	R,R	3
08	FEET		1	28				48				68	GET	R,R	3
09	METERS		1	29				49				69	PUT	R,R	3
1A	MILLISEC		1	2A				4A				6A			
0B	SECONDS		1	2B				4B				6B			
0C	HOURS		1	2C				4C				6C			
0D				2D				5D	INCRMENT	R	2	6D			
0E	REPEAT		1	2E				4E	DECRMENT	R	2	6E			
0F	RETURN		1	2F				4F	CLEAR	R	2	6F			
10	FORWARD	R*	2	30	FORWARD	N*	3	50	JUMP	N	3	70	SET	N,R	4
11	REVERSE	R	2	31	REVERSE	N	3	51	JUMPEQ	N	3	71	ADD	N,R	4
12	RIGHT	R	2	32	RIGHT	N	3	52	JUMPNE	N	3	72	SUB	N,R	4
13	LEFT	R	2	33	LEFT	N	3	53				73	MUL	N,R	4
14	RTURN	R	2	34	RTURN	N	3	54	SBROUTNE	N	3	74	DIV	N,R	4
15	LTURN	R	2	35	LTURN	N	3	55	ARRAY	N	3	75			
16				36				56				76			
17	BEEP	R	2	37	BEEP	N	3	57				77	COMPARE	N,R	4
18	DISPLAY	R	2	38	DISPLAY	N	3	58				78	GET	N,R	4
19	HESITATE	R	2	39	HESITATE	N	3	59				79	PUT	N,R	4
1A				3A	CALL	N	3	5A				7A			
1B	TEST	R	2	3B				5B				7B			
1C	ENTER	R	3	3C				5C				7C			
1D	RANDOM	R	2	3D				5D				7D			
1E	SCAN	R	2	3E				5E				7E			
1F	PROXIMITY	R	2	3F				5F				7F			

Table 2: *Pseudoprocessor instruction set (*R = Register, N = Number, RC = Register Count).*

SCOTCH DISKETTES at 50% off!

ALL 3-M MEDIA AVAILABLE at 50% or MORE OFF LIST PRICE

Why settle for inferior substitutes when you can now get the very best media at unprecedented savings. Be it 8" or 5¼", hard- or soft-sectored: We are an Authorized 3-M Distributor and can ship it immediately.

Examples:

Part #		List	BHRT PRICE
744D	5¼" SSDD (hard/soft sector)	4.20	$2.04
745D	5¼" DSDD (hard/soft sector)	6.00	2.94
741D	8" SSDD (hard/soft sector)	5.55	2.75
743D	8" DSDD (hard/soft sector)	7.10	3.39

(per diskette, in box of 10, individual envelopes)

(even greater savings on diskettes in bulk pack—call for information)
Minimum order: 10 boxes. For orders of 1000 or more deduct 2%, for 5000 or more deduct 5%, for 10,000 or more deduct 7.5%. Call for savings on larger orders.

3M Authorized Distributor

BHRT

ORDERS & INFORMATION:
Mo.-Fri. 9:00 AM-5:30 PM PST, Sat. 12:00 AM-5:00 PM PST
1-(800) [illegible] CA, AK, HI call (714) 783-1363

BANK REFERENCE: BARCLAYS BANK OF CALIFORNIA (213) [illegible]
AMERICAN EXPRESS, VISA, MASTERCHARGE
APO, FPO, INTERNATIONAL ORDERS ACCEPTED
INC. P.O. Box 3791, Riverside, CA 92519

Terms: Cashier's check, Visa, MasterCharge, American Express (add 4% for credit cards). Purchase orders from approved organizations on 30 day net at 2% surcharge. COD at 10% surcharge. Shipping charges extra.

Circle 508 on inquiry card.

BEST COPY AVAILABLE

devices and place the result into the specified register.

The next group of instructions, 20 through 2F hexadecimal, cause their respective motions until the proximity register changes; then motion ceases. The elapsed distance units are then recorded in the given register.

Instructions 30 through 3F hexadecimal are 3-byte commands that use an immediate 16-bit constant for the operand. For example, BEEP 7 (37 0007) signals the robot to beep seven times. In other respects, this group mirrors the group 10 hexadecimal instructions, except for the CALL instruction; CALL is a "hook" to an assembly-language subroutine, and its operand is a 16-bit absolute address.

Five instructions work with the processor registers. JUMP A takes the signed 16-bit value in register A and adds it to the program counter. JUMPEQ and JUMPNE are conditional jumps that test the condition code register. SBROUTNE is a simple subroutine call to a section of the program, and it terminates with a RETURN instruction. Each processor register instruction uses an operand that is a relative offset to the program counter so that the resulting code is position-independent.

The ARRAY instruction declares the array being referenced by adding its operand to the program counter and placing the resulting address in the array register. The INCRMENT, DECRMENT, and CLEAR instructions are added for convenience.

GET and PUT are used with the ARRAY instruction. After an array has been declared, a GET 5,B (70 0005 02) reads the fifth 16-bit element from the array and stores it in register B. The PUT A,B instruction (69 01 02) uses the number in register A as the offset into the array and places the number found there in register B.

Finally, the register-manipulation instructions use a constant or a register to perform an operation on another register. For instance, ADD B,C (60 02 03) means "add register B to register C." COMPARE compares a constant or a register to another register and then sets the appropriate condition code.

Some Examples

Armed with these instructions, let's try them on a simple program. Suppose we want our simple robot to search for a source of light in the room, determine how far away it is, and then report back to us. The resulting program might look something like listing 2. In this program, the robot turns to the right until it faces a source of light. Then it moves forward until it senses or bumps into the source of light, whereupon it moves back to its starting position, displays the distance, and stops. This program requires only 17 bytes through the keypad.

Of course, that is assuming the light sensor is mounted on the front of the robot and the path to the light source is unobstructed. If the program has to take obstructions into account, a triangulation scheme could be worked into it (but that, as they say, is left "as an exercise for the

★ ★ ★ FEATURING 8 AND 16 BIT SYSTEMS ★ ★ ★

EPSON QX VALDOCS: Extremely user friendly! See review Sept. BYTE.
HX 80: Notebook-sized battery operated Z 80 computer, up to 256 K RAM, built-in hard copy, LCD scrollable screen.

MASTER MAX: S-100 system, Z-80, INTERCONTINENTAL CPZ48000 single card computer with four channels of DMA, dual 8" double density drives, CP/M .... $2,540.
Options: double sided drives, Winchester, TURBODOS, 2 user, 220v/50hz.

IMS 8X MULTIUSER SYSTEMS: Z-80, S-100. Each user has own Z-80, 64K RAM, 2 I/O. TURBODOS multiuser CP/M compatible operating system cuts link/edit time in half. Z-80 code. Interrupt driven. 8088 upgrade w/256K RAM has been announced.

TARBELL: Empire systems, Z-80, S-100.
CROMEMCO: C-10 personal computer w/software package .... $1,695.

8088: COLUMBIA DATA: IBM-PC look alike, multiuser option.

8086 S-100 SYSTEMS:
LOMAS: with MS-DOS or CP/M-86. Winchester option.
SEATTLE: with simultaneous 8" and 5" drives. Will accept IBM/PC software.

DUAL PROCESSOR SYSTEMS:
GODBOUT 816 A,B,C: 8085/8088. MP/M 816 allows simultaneous operation of both processors.
CROMEMCO DPU: 68000 and Z 80. CROMIX operating system.

MAX BOX 8" DRIVE SUBSYSTEMS w/QUMES, SHUGARTS, MITSUBISHI, NEC.

PRINTERS (dot matrix and LQ):
EPSON, NEC, QUME, C.ITOH, IDS, FLORIDA DATA, TELETYPE.

TERMINALS: WYSE, HAZELTINE, IBM 3101, TELEVIDEO.
Voice recognition board for TELEVIDEO 950.

PER SCI 277/299 DRIVES.

MODEMS: U.S. ROBOTICS 1200/300 DC HAYES compatible .... $525.

S-100 MAINFRAMES:
PARADYNAMICS, ECT, some TEI 12 slot still in stock.

GRAPHICS:
MICROANGELO GRAPHICS.
MIRAGE: new from SCION.
AUTO-CAD Interactive graphics software; for engineers, architects, designers.
HOUSTON INSTRUMENTS PLOTTERS, DMP-29 .... $1,775.

IBM PC ACCESSORIES: Extensive line including QUADRAM, SEATTLE, 8080/8086 EMULATOR (software).

IBM 3270: compatible equipment from Teletype Corp. Fast delivery! Cost effective!

We have an extensive product line including systems, peripherals, software, boards, drives, consulting services. Write or call for detailed specifications. We have knowledgeable technical staff.

WE EXPORT Overseas Callers Phone (212) 448-6298
TWX 710 588 2844 or Cable OWENSASSOC

JOHN D.OWENS Associates, Inc.
12 Schubert Street, Staten Island, New York 10305
(212) 448-6283 (212) 448-2913 (212) 448-6298

BEST COPY AVAILABLE

BEST COPY AVAILABLE

Listing 2: Sample program written in ROTERP. The robot is commanded to scan the room for a light source, determine its distance, and display the information.

```
0000    12    0001    START    RIGHT             1
0003    2B    01               SCAN              A
0005    2E    01               TEST              A
0007    42    FFF8             JUMPEQ        START
0010    30    02               COUNT FORWARD     B
0012    21    02               REVERSE           B
0014    2B    02               DISPLAY           B
0016    00                     HALT
```

reader"). Imagine how much more difficult it would be in assembly language!

Let's try programming the robot to survey a room. An array can be declared to store distances and positions, data that can be acquired during the robot's trips back and forth throughout the room until it is familiar with its surroundings.

Even this simple robot can be taught to wake you when the sun comes up. Just give it explicit instructions to go to an eastern bedroom window and then let the program loop patiently until light appears. At dawn the robot will retrace its steps to your bed and beep insistently.

The Implementation

The implementation of ROTERP is easier than I would have expected. This version fits into a single 2K-byte 2716 EPROM (erasable programmable read-only memory). It is also fairly memory-efficient: the memory required for the registers, temporary locations, and a 10-deep nested call stack is only 128 bytes, leaving the rest of available memory for program storage. (I used a Motorola MEK 6800 D2 microprocessor kit to arrive at these figures.)

Figure 2 shows the modules that need to be written to form the complete interpreter. The first module allocates some memory for all of the 16-bit processor registers, some temporary storage for use by the interpreter, and enough memory for a subroutine call stack, which must accommodate as many nested calls as might be reasonably needed. You can use 1 byte to allocate as many as 256 general-purpose registers. The tradeoff is simply between memory used by ROTERP and the amount of memory left for program storage.

The initialization module is entered after the program is placed into memory. First it places the fixed starting address of the program into the program counter, sets up any interrupt-service-routine vectors that are needed, and initializes any device that needs it. Next it calls a subroutine that clears all of the general-purpose registers and places some default values into the special-purpose reg-

Martin Marietta Aerospace

Data Processing Opportunities

Martin Marietta Aerospace, NASA's prime Contractor for the Space Shuttle External Tank has immediate openings for Data Processing professionals. Because we actually manufacture the external tank, you'll get to see the actual results of your efforts.

COMPUTER PROGRAMMER/ ANALYSTS

Immediate opportunities exist for individuals experienced in:

- **UNIVAC 1100**
  ASC11 COBOL
  DMS 1100
  DDL, SDDL, DMU
  DML, QLP
  DPS 1100, TIP
  D/B Editor

- **APPLICATION EXPERIENCE**
  Shop floor control, Scheduling, Manufacturing, Inventory, Purchasing, Configuration Management, Quality Engineering.

- **DATA BASE OPENINGS**
  Analyst, Design, Administrators with above hardware, software and applications experience.

These opportunities exist at our Michoud Assembly Facility located in suburban East New Orleans.

Qualified candidates interested in learning more about these opportunities at Martin Marietta should forward resumes, including salary history to Martin Marietta Aerospace, Denver Glazier, BYTE-383, P.O. Box 29304, New Orleans, Louisiana 70189. We are an equal opportunity employer, m/f/h.

MARTIN MARIETTA

Everybody's Logic Analyzer

12 Channels
16 Words

A logic probe and oscilloscope are no longer adequate for analysis in today's digital world. For testing or debugging microcomputer or other digital logic circuits you need a real logic analyzer.

The LA-12 captures, stores and displays TTL and LSTTL digital data so that the instantaneous meaning of the data stream (e.g. data value, ASCII code, address) can be understood and analyzed long after the actual events have passed.

▪ Easy to Use ▪ 10 MHz ▪ Clock Qualifier ▪ Trigger input ▪ 3 Trigger Qualifiers ▪ Built-in LED Display — No oscilloscope needed ▪ Compact ▪ Expandable ▪ Low Cost

30 day trial
Purchase an LA-12, use it, and if you are not completely satisfied, return it within 30 days and receive a full refund.

Free Offer
If you order within 45 days, *and mention this magazine*, you will receive a $49.95 input cable free with each LA-12 ordered.

Save $28.95
In addition, if you enclose payment with your order you can deduct 5% and we will pay shipping charges.

All prices are in US dollars for 120VAC.

To order in the Continental US call
TOLL FREE
1-(800) 547-5995 EXT. 195

Connecticut microComputer, Inc.
36 Del Mar Drive, Brookfield, CT 06804
(203) 775-4595 TWX: 710-456-0052

Q	Description	Price	Total
	Logic Analyzer	$379.00	
	Input Cable	49.95	
	20 Color-coded microclips	44.95	
Connecticut residents add 7½% sales tax			
	Shipping & Handling	$10.00	
	Total		

☐ Company purchase order enclosed (Rated Firms only)
☐ Check ☐ VISA ☐ MasterCard
Acct. No. ____________
Signature ____________ Exp. Date ____
Name (Print) ____________
Address ____________
____
City ____________
State ____________ Zip ____

Dealer inquiries invited

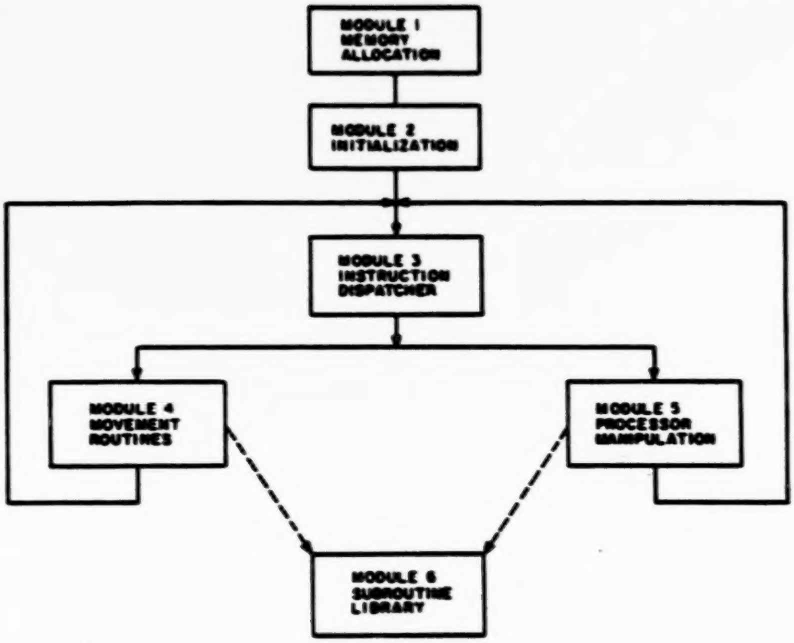

MODULE 1 MEMORY ALLOCATION
MODULE 2 INITIALIZATION
MODULE 3 INSTRUCTION DISPATCHER
MODULE 4 MOVEMENT ROUTINES
MODULE 5 PROCESSOR MANIPULATION
MODULE 6 SUBROUTINE LIBRARY

Figure 2: *Block diagram of ROTERP modules.*

isters. ROTERP uses SECONDS, FEET, and SLOW as defaults. This "clear and default" function is expressed as a subroutine because it is also called by the ZERO instruction.

Execution then continues into the instruction dispatch module, which contains a jump table used to pass control to the routine that executes the current instruction. If any invalid op code is executed, control is transferred to a routine that displays an error code and jumps the SBC monitor. This jump table is 256 bytes long because all op codes greater than 7F are currently treated as invalid instructions. If you want all 256 possible instructions, the jump table size would increase to 512 bytes.

The next section, which contains all the routines that carry out each instruction, is divided into two modules: one for execution of movement instructions and one for register-manipulation instructions. The instruction routines in the movement module are very dependent on the hardware scheme that controls the direction and speed of the motors. The motors can be interrupt-driven or controlled with timing loops in the manner of stepping motors and simple relay-operated DC motors.

All of the movement routines have many functions in common. First, they get the operands, translate them from a register designation to a working number as needed, and assign a temporary location to each working number. A scaling routine then translates the number of distance or time units into an absolute number expected by the motor handler. The op code itself serves as a direction indicator.

Next, the motor handler is called to perform the motion. And the last function every instruction performs is to add the length of the instruction to the program counter so that it points to the next instruction to be executed. Control is then passed back to the instruction dispatcher.

Because many of the instructions must perform common functions, the functions are expressed as subroutines and placed in the last module, the subroutine library. The subroutine library also includes routines that do common register operations like 16-bit addition, subtraction, multiplication, and division. The library is designed not only to save duplication of code but to permit the easy addition of new instructions.

To add a new instruction, you

BEST COPY

simply place the new entry in the jump table and call the appropriate "fetch operand" routines. If the instruction is in the register-manipulation group, you can call many of the existing subroutines in the library to reduce the amount of new code that needs to be written. (If the instruction involves the use of an entirely new device, you must write a handler for that device, as you would in any case.) Finally, an existing routine increments the program counter and jumps back to the instruction dispatcher.

Extensions

If you find yourself using a particular sequence of instructions repeatedly, a single new instruction will make your program both shorter and easier to write. And the processor is not limited to expanding new instructions: a new device may be complex enough to warrant the use of a new special-purpose register. If you expect to use many different arrays, adding new array registers is only a matter of allocating space for them and assigning new op codes that call the same subroutines that the present ARRAY instruction uses.

The simple architecture and instruction set presented here should not be considered complete or taken for the optimum design of a pseudoprocessor for robots. Each design should be tailored individually to the kinds of experiments that you expect to do. For example, if you want a link to a host computer, the instruction dispatch module can be modified to receive instructions from a serial-line interface instead of reading them from memory. Two other options exist: the processor can be oriented around 32-bit registers if more precision is desired, or it can use 8-bit registers if less precision is needed. Expressing movement routines in terms of relative polar or Cartesian coordinates might be more desirable.

For maximum flexibility, one interpreter could include all of these features. If one byte does not sufficiently represent all of the different instructions, a 16-bit op code could be used. Better yet, a single instruction that changes jump tables would allow you to switch between whole instruction sets. The ultimate step forward, though, would be a translator that would produce running code from the instruction mnemonics. This translator would resolve symbolic references and might be the precursor of even higher-level "macro" statements of behavior. New language constructs, such as those found in LISP, might be implemented for ease of integrating the work being done in other areas of artificial intelligence.

Summary

ROTERP illustrates how a pseudoprocessor can control the increasing number of devices that can be incorporated into an experimental robot. With careful planning, you will be able to improve and expand the interpreter without difficulty. By tailoring the robot's processor to the kind of behavior it will perform, the robot becomes a much easier tool to work with.■

ALL 1/3 off SOFTWARE

Software Catalog. FREE. Get the lowest price on *every* software program you need. 1/3 off list price *guaranteed*. And look what else you get from ITM.

Unlimited Consultation! ITM's consultants work with an amazing database indexed with over 300 software selection criteria – plus thousands of in-depth product reviews. You'll get the programs that are right for you. Quickly. Easily. It's the most sophisticated consultation service in the industry. Call toll free!

Over 2,000 programs to choose from. Hundreds added every month. All categories. All formats.

See any program demonstrated. Order any non-entertainment program for *a risk-free, thirty-day trial*. Full refund if not completely satisfied.

No minimum order. Buy just one program if you like. Quick shipments.

All this, and more, is yours for an annual fee of $100. Call ITM now. Or, mail the coupon today and receive our FREE Software Catalog. Save hundreds – even thousands – of dollars on all of your software purchases with ITM.

American Express, Visa, or MasterCard honored.
Call toll-free today.
(800) 334-3404
In California (415) 284-7540

Software Catalog. FREE. Byte 383

☐ YES. Send complete information for my review. Please include a free copy of ITM's Software Catalog.

NAME____________
COMPANY NAME____________
ADDRESS____________
CITY/STATE/ZIP____________
PHONE____________
☐ Individual ☐ Dealer ☐ Consultant ☐ Company

ITM
Software Division
We make software buying simple.

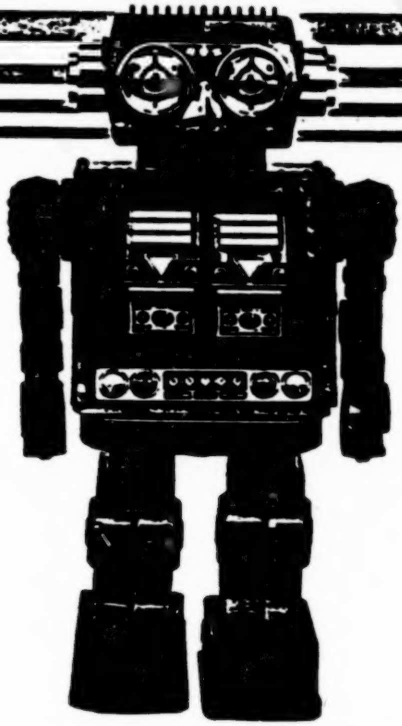
Attn. Steven Cloudtree
936 Dewing Ave., Suite E
Lafayette, CA 94549-4292
(800) 334-3404 or
(415) 284-7540

Circle 204 on inquiry card.

BEST COPY AVAILABLE

BEST COPY AVAILABLE

How NOT to buy a robot



Your mission: Bring in a robot. Here are 10 errors you'll want to avoid

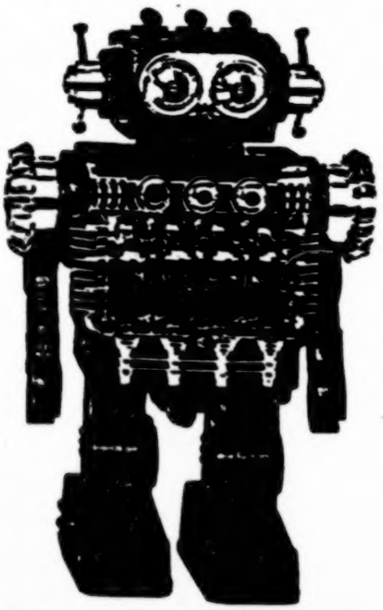
By Jack Thornton

The plant manager called you into his office today—a blue suit, white shirt event because you know the board of directors was in to see him yesterday. And before you even hear it, you know what the message is: the Big Guys were appalled that Good Old Plant 27 is (1) slipping into the red and (2) has no robots.

By the time you come out of the boss's office (1:30 p.m. and still no lunch) you have a project: acquiring your company's first robot. You also have a three-month deadline, lots of high-level attention and no idea where to start.

Every manager ought to have his own organizing methods well thought-out, even down to the unknowns that always trip up other managers. But what about the unknown unknowns ("unk-unks," in aerospace slang)? Where do you look for pitfalls?

Here are 10 that could ruin your first robot purchase:



1 Buy a robot solely to replace a high-seniority, highly skilled (and therefore costly) worker on a one-for-one basis. Preferably a man or woman with clout in the union so you can show them (and management) you mean business.

2 Keep the robot's purchase—let alone its estimated time of arrival—a secret, especially from the union. Competitors might get wind of it.

3 Insist that the robot move 500-lb. workpieces at 400 inches a minute. And that it set them down within plus-or-minus 0.002 inch. That way you can get rid of the forklift and the jib cranes, too.

4 Buy it from your brother-in-law who sells your company janitorial supplies.

5 Ignore complex high-tech talk like interface, orientation, dunnage, end effector and part design. These are just worrybead words to enrich consultants.

6 Don't ask bidders for names of their customers who've tackled jobs similar to yours. Your idea is too unique to share. They're too secretive to help, anyway.

7 Promise your boss that robots can be cost-justified as simply and rationally as any machine tool.

8 Pass along as Gospel to your boss (and his boss) vendor promises of startup and operating uptime.

9 Take the low bid, no matter what.

10 Don't bother with after-startup record keeping. You might get trapped by your own paperwork.

BEST COPY AVAILABLE

All of those well-trod paths have led to robotic flops. Any reasonably experienced and responsible robot vendor would, in fact, call you instantly on almost any of these points. No one wants to forego repeat sales. But the reasons why these actions will be troublesome are not always obvious. In the course of exploring them, we'll get a good idea of what industrial robots are and are not.

If you have a union (Item 1) its leadership is undoubtedly aware of robot possibilities. And possibly fearful. If you ask, they may show you a job that would be well worth automating. And they may not, of course. But pick a sure-fire winner. Keep it simple. Provoking the union, intentionally or otherwise, is foolish. Sabotage, though little discussed, does happen.

This leads into Item 2. Your own union tradesmen probably will have to install your robot and possibly even teach-program it. Moreover, this complex piece of machinery must mesh perfectly with previously installed equipment. Everyone involved should be alerted. You can't start on these problems too soon.

You should be aware that setting up a robot as a display to familiarize workers doesn't always work. They may ignore it until they realize it's going into their department.

Job specifications like those in Item 3 are beyond the capability of all but the most sophisticated gantry robots: $500,000 and up. That would pay a lot of forklift operators. Robots don't work faster than a well-motivated human anyway—at least not yet. Robots' virtues are their tirelessness, their inability to file workers' comp claims, to go on strike, or to quit and go to work for a competitor.

They do, however, get sick. Two percent of the time—minimum. That's the best results reported from General Motors, the nation's most experienced robot user. Plan on 10 percent downtime at first.

Choosing the right robot vendor (Item 4) is critical. There's a big argument in the industry now about buying from robot builders vs. systems houses vs. your favorite machine tool dealer. Each has a strength and a corresponding weakness:

- Builders only want to sell their own hardware. But they may well have applied their robot to your job dozens of times. The experience they have is priceless.
- Systems houses pick the best robot for the job. But only a handful have real depth of experience. Robot builders, at least, are less than thrilled by their performance.
- Machine tool distributors (and tool builders) are rather new at robotics (except for Cincinnati Milacron). But they will know your machinery and your plant as no systems house or robot maker could.

Understanding buzzwords in Item 5 is crucial to your success. Robots themselves do nothing. They must be married up to other equipment, generally things that were not built with robots in mind. Your in-house engineering and tradespeople have to do this.

Orientation of parts reaching the robot is very important: robots are virtually blind. Only human workers have two hands and arms. Robots don't. Only costly and risky specials have a second arm to handle dunnage and packing materials.

End effectors—grippers and hands—are analogous to the cutting tools on your lathes and machining centers, but more complex. Cutting tools are standardized in packages for your convenience. You'll have to precisely specify robot tooling. Don't underestimate the problem.

Part redesign is commonplace in robotics. As savvy a welder and automation user as Deere & Co. had to redesign its farm tractor cabs a few years back when it shifted to robot arc welding. The existing design thwarted the robots.

Robots don't have thumbs. Their arms and hands have less than 10 percent of the mobility of human arms and hands. Don't expect them to handle screws and springs. Avoid asymmetrical parts, or parts which must be fixtured. Avoid subassemblies with lots of little parts.

Always ask vendors for names of customers (Item 6). Some robots simply won't do some jobs. There is not now—and never was—any substitute for first-hand experience. The time you spend finding someone "who's been there before you" may be the most valuable time you'll spend in the entire project.

Robots cannot be justified like a machine tool (Item 7). Machine tools do specific operations on certain types of parts within certain sizes, certain materials, etc. But a robot machine loader can become part of a material handling or spray painting system if job-needs change.

Remember that while a robot can work the second and third shifts virtually without extra costs, you will lose, if it must do so to be justifiable. Unless, of course, your justification includes the costs of everything else the robot impacts in the plant.

Robots must partly be justified on avoidance of certain costs. Many of these are obviously labor-related. But many are hidden: warranty repairs, finished-goods scrap, exposure to Environmental Protection Agency and Occupational Safety and Health Administration dicta, field service calls on improperly assembled (or welded or mishandled) parts.

Item 8: some robot vendors will make dramatic promises of speedy startups and 99-percent-plus uptime, qualifying those mildly with something like "as long as you do your part." Quite a loophole if this is your first robot. The industry has service problems. There is a shortage of skilled field installers, servicemen and troubleshooters. The best of them are quickly hired by other customers. They try, but...

Taking the low vendor bid (Item 9) is a double fallacy. First, the robot itself is just a third or a fourth of the total capital you're going to spend. Second, blindly taking the low bid discounts all the study you've done on vendors.

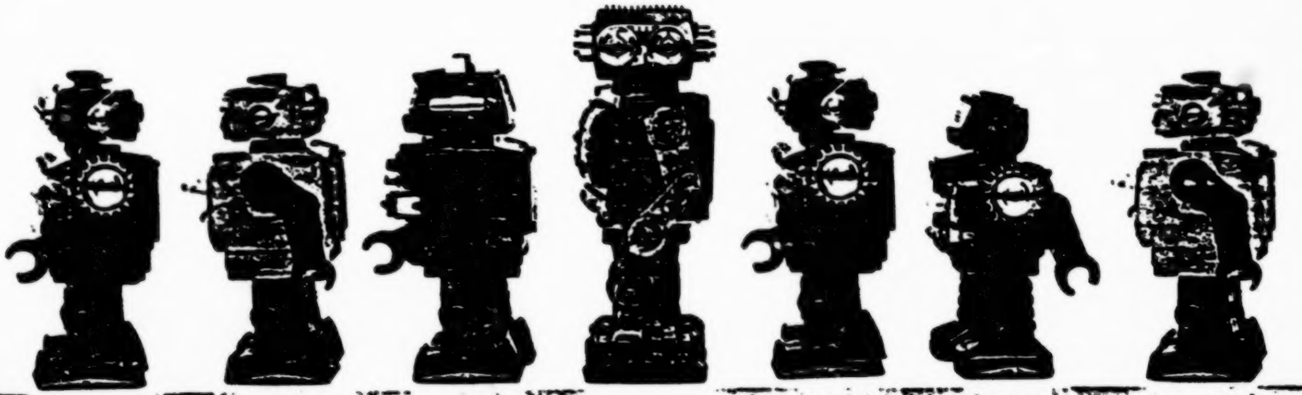
Record keeping (Item 10) may save your job if problems occur. At a minimum, records will justify additional projects later. Moreover, when a manufacturing system "goes down," the robot tends to get the blame. It's the most visible target: high-tech's high profile. After that, you are probably the second most visible.

None of the foregoing was meant to tell you how to choose between electric, hydraulic and pneumatic robots or how many axes to buy (two to seven.) Those are engineering decisions.

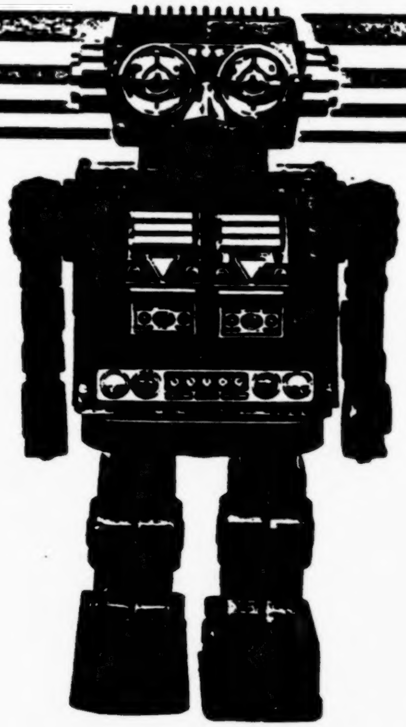
Some over-simplified axioms: electrics are most accurate and best at standing still if that's what the job calls for. Pneumatics are best for small, snappy and not-too-precise jobs. Hydraulics have the brute force but are sometimes slow and may leak. Since they are built from high-quality shock absorbers, they are the least accurate.

Finally, this is not an attempt to define a formula for return on investment. If the robot industry agrees on anything it's that business-school-style ROI calculations are always misapplied.

Jack Thornton is editor of Robot Insider, a weekly newsletter published by Fairchild Publications.



BEST COPY AVAILABLE



How NOT to buy a robot

Your mission: Bring in a robot. Here are 10 errors you'll want to avoid

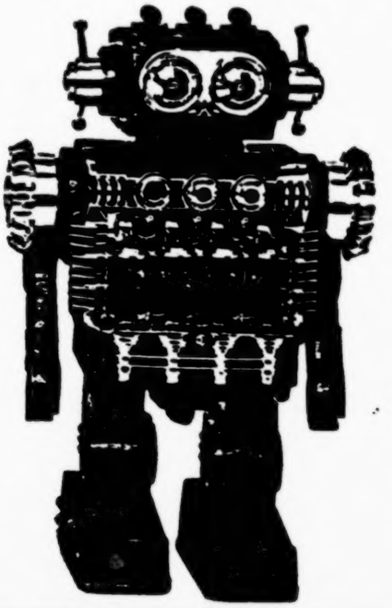
By Jack Thornton

The plant manager called you into his office today—a blue suit, white shirt event because you know the board of directors was in to see him yesterday. And before you even hear it, you know what the message is: the Big Guys were appalled that Good Old Plant 27 is (1) slipping into the red and (2) has no robots.

By the time you come out of the boss's office (1:30 p.m. and still no lunch) you have a project: acquiring your company's first robot. You also have a three-month deadline, lots of high-level attention and no idea where to start.

Every manager ought to have his own organizing methods well thought-out, even down to the unknowns that always trip up other managers. But what about the unknown unknowns ("unk-unks," in aerospace slang)? Where do you look for pitfalls?

Here are 10 that could ruin your first robot purchase:



1 Buy a robot solely to replace a high-seniority, highly skilled (and therefore costly) worker on a one-for-one basis. Preferably a man or woman with clout in the union so you can show them (and management) you mean business.

2 Keep the robot's purchase—let alone its estimated time of arrival—a secret, especially from the union. Competitors might get wind of it.

3 Insist that the robot move 500-lb. workpieces at 400 inches a minute. And that it set them down within plus-or-minus 0.002 inch. That way you can get rid of the forklift and the jib cranes, too.

4 Buy it from your brother-in-law who sells your company janitorial supplies.

5 Ignore complex high-tech talk like interface, orientation, dunnage, end effector and part design. These are just worrybead words to enrich consultants.

6 Don't ask bidders for names of their customers who've tackled jobs similar to yours. Your idea is too unique to share. They're too secretive to help, anyway.

7 Promise your boss that robots can be cost-justified as simply and rationally as any machine tool.

8 Pass along as Gospel to your boss (and his boss) vendor promises of startup and operating uptime.

9 Take the low bid, no matter what.

10 Don't bother with after-startup record keeping. You might get trapped by your own paperwork.

All of those well-trod paths have led to robotic flops. Any reasonably experienced and responsible robot vendor would, in fact, call you instantly on almost any of these points. No one wants to forego repeat sales. But the reasons why these actions will be troublesome are not always obvious. In the course of exploring them, we'll get a good idea of what industrial robots are and are not.

If you have a union (Item 1) its leadership is undoubtedly aware of robot possibilities. And possibly fearful. If you ask, they may show you a job that would be well worth automating. And they may not, of course. But pick a sure-fire winner. Keep it simple. Provoking the union, intentionally or otherwise, is foolish. Sabotage, though little discussed, does happen.

This leads into Item 2. Your own union tradesmen probably will have to install your robot and possibly even teach-program it. Moreover, this complex piece of machinery must mesh perfectly with previously installed equipment. Everyone involved should be alerted. You can't start on these problems too soon.

You should be aware that setting up a robot as a display to familiarize workers doesn't always work. They may ignore it until they realize it's going into their department.

Job specifications like those in Item 3 are beyond the capability of all but the most sophisticated gantry robots: $500,000 and up. That would pay a lot of forklift operators. Robots don't work faster than a well-motivated human anyway—at least not yet. Robots' virtues are their tirelessness, their inability to file workers' comp claims, to go on strike, or to quit and go to work for a competitor.

They do, however, get sick. Two percent of the time—minimum. That's the best results reported from General Motors, the nation's most experienced robot user. Plan on 10 percent downtime at first.

Choosing the right robot vendor (Item 4) is critical. There's a big argument in the industry now about buying from robot builders vs. systems houses vs. your favorite machine tool dealer. Each has a strength and a corresponding weakness:

- Builders only want to sell their own hardware. But they may well have applied their robot to your job dozens of times. The experience they have is priceless.
- Systems houses pick the best robot for the job. But only a handful have real depth of experience. Robot builders, at least, are less than thrilled by their performance.
- Machine tool distributors (and tool builders) are rather new at robotics (except for Cincinnati Milacron). But they will know your machinery and your plant as no systems house or robot maker could.

Understanding buzzwords in Item 5 is crucial to your success. Robots themselves do nothing. They must be married up to other equipment, generally things that were not built with robots in mind. Your in-house engineering and tradespeople have to do this.

Orientation of parts reaching the robot is very important: robots are virtually blind. Only human workers have two hands and arms. Robots don't. Only costly and risky specials have a second arm to handle dunnage and packing materials.

End effectors—grippers and hands—are analogous to the cutting tools on your lathes and machining centers, but more complex. Cutting tools are standardized in packages for your convenience. You'll have to precisely specify robot tooling. Don't underestimate the problem.

Part redesign is commonplace in robotics. As savvy a welder and automation user as Deere & Co. had to redesign its farm tractor cabs a few years back when it shifted to robot arc welding. The existing design thwarted the robots.

Robots don't have thumbs. Their arms and hands have less than 10 percent of the mobility of human arms and hands. Don't expect them to handle screws and springs. Avoid asymmetrical parts, or parts which must be fixtured. Avoid subassemblies with lots of little parts.

Always ask vendors for names of customers (Item 6). Some robots simply won't do some jobs. There is not now—and never was—any substitute for first-hand experience. The time you spend finding someone "who's been there before you" may be the most valuable time you'll spend in the entire project.

Robots cannot be justified like a machine tool (Item 7). Machine tools do specific operations on certain types of parts within certain sizes, certain materials, etc. But a robot machine loader can become part of a material handling or spray painting system if job-needs change.

Remember that while a robot can work the second and third shifts virtually without extra costs, you will lose, if it must do so to be justifiable. Unless, of course, your justification includes the costs of everything else the robot impacts in the plant.

Robots must partly be justified on avoidance of certain costs. Many of these are obviously labor-related. But many are hidden: warranty repairs, finished-goods scrap, exposure to Environmental Protection Agency and Occupational Safety and Health Administration dicta, field service calls on improperly assembled (or welded or mishandled) parts.

Item 8: some robot vendors will make dramatic promises of speedy startups and 99-percent-plus uptime, qualifying those mildly with something like "as long as you do your part." Quite a loophole if this is your first robot. The industry has service problems. There is a shortage of skilled field installers, servicemen and troubleshooters. The best of them are quickly hired by other customers. They try, but...

Taking the low vendor bid (Item 9) is a double fallacy. First, the robot itself is just a third or a fourth of the total capital you're going to spend. Second, blindly taking the low bid discounts all the study you've done on vendors.

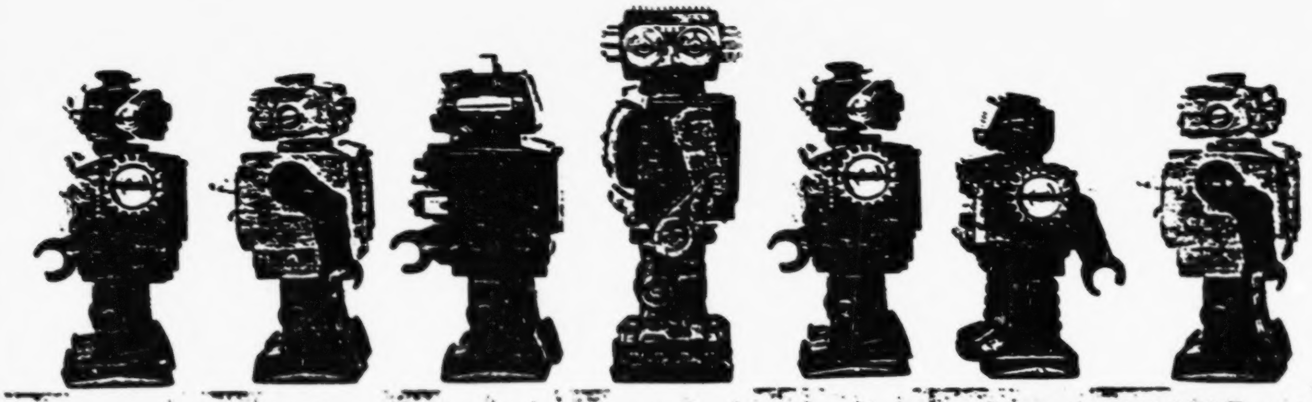
Record keeping (Item 10) may save your job if problems occur. At a minimum, records will justify additional projects later. Moreover, when a manufacturing system "goes down," the robot tends to get the blame. It's the most visible target: high-tech's high profile. After that, you are probably the second most visible.

None of the foregoing was meant to tell you how to choose between electric, hydraulic and pneumatic robots or how many axes to buy (two to seven.) Those are engineering decisions.

Some over-simplified axioms: electrics are most accurate and best at standing still if that's what the job calls for. Pneumatics are best for small, snappy and not-too-precise jobs. Hydraulics have the brute force but are sometimes slow and may leak. Since they are built from high-quality shock absorbers, they are the least accurate.

Finally, this is not an attempt to define a formula for return on investment. If the robot industry agrees on anything it's that business-school-style ROI calculations are always misapplied. AMM

Jack Thornton is editor of Robot Insider, a weekly newsletter published by Fairchild Publications.



BEST COPY AVAILABLE